College of Computer Sciences & Engineering

Computer Engineering Department

COE 560
Computer Communication Networks

Pre-Midterm

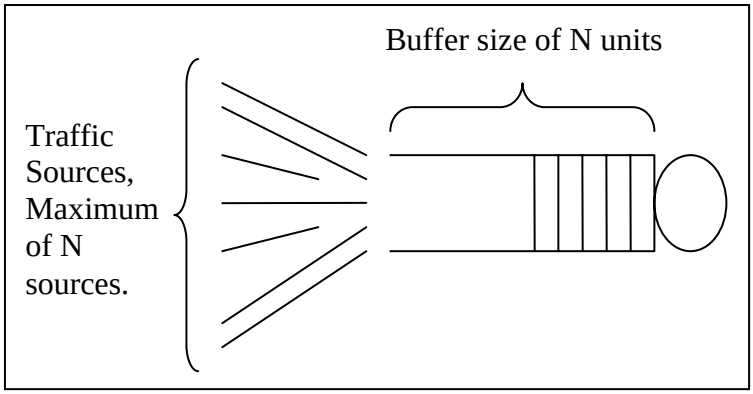
Instructor: Dr. Abdulaziz Al-Mulhem

Submitted by:

Bambang Ali B. Sarif
Yousif E. Al-Dilaijan
Garba Isa Yau
Khalid Al-Badawi
Mohammed Faheemuddin
Mohammed Riyazuddin
Mohammed Yahya Imam
Naser S. Al-Abeedi
Syed Sanaullah

Due Date: Wednesday, 17th April 2002.

Question:



Find P_{ij} : the probability of having i packets in the buffer and j active sources.

Solution:

In order to find the probability distribution of the buffer occupancy, consider the following 2-D state diagram.

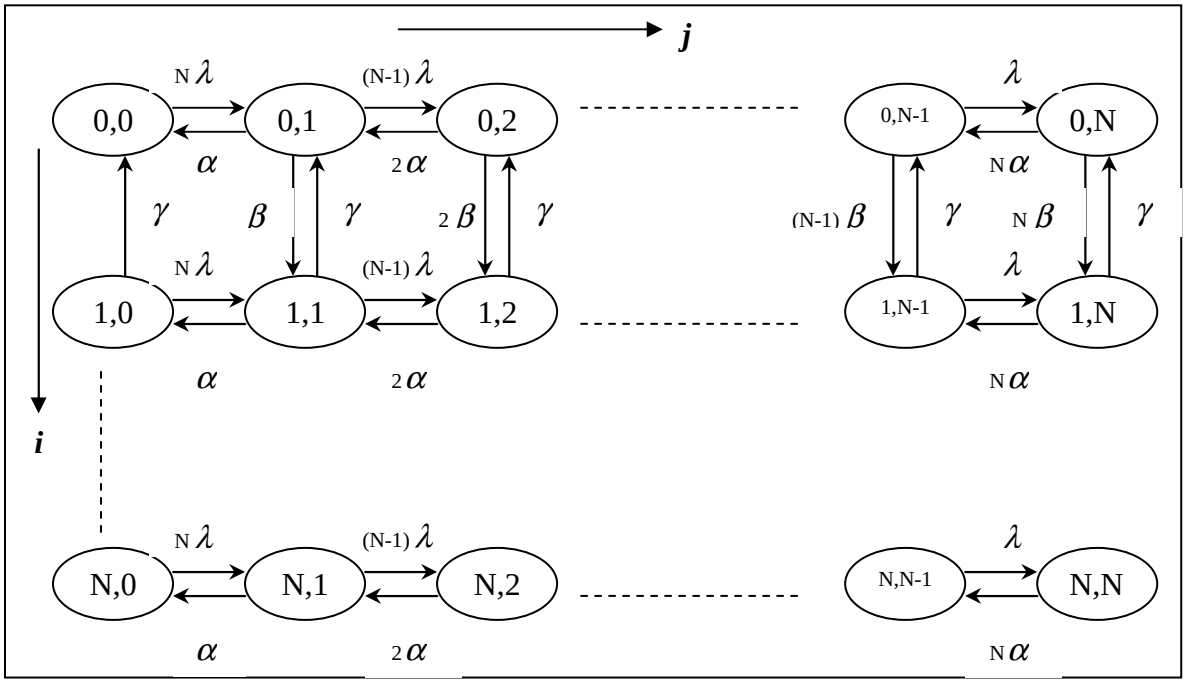


Figure 2: state-space transition diagram of the problem

We want to solve for P_{ij} using the 2-D state space, using algorithmic technique.

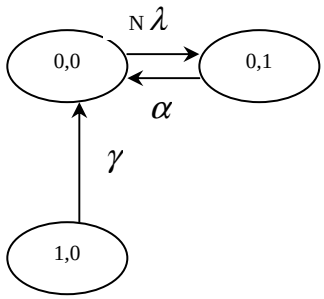
We can differentiate 2 parts in state-space transition diagram:

- for i = 0
- for i > 0,

and for each row we have 3 different cases: j = 0, 1≤ j < N, j = N.

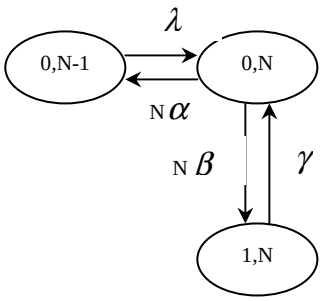
First element (j=0)

$$N\lambda P_{00} = \alpha P_{01} + \gamma P_{10}$$
$$\Rightarrow P_{00} = (1 - N\lambda)P_{00} + \alpha P_{01} + \gamma P_{10}$$



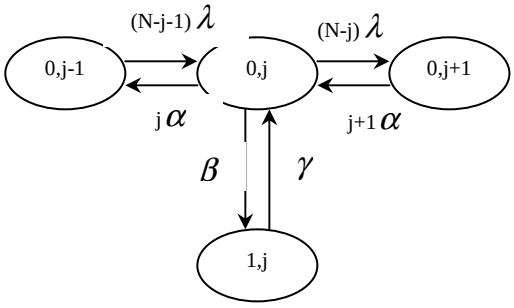
Last element (j=N)

$$P_{0N}[N\alpha + N\beta] = \lambda P_{0,N-1} + \gamma P_{1N}$$
$$\Rightarrow P_{0N} = \lambda P_{0,N-1} + [1 - N\alpha - N\beta]P_{0N} + \gamma P_{1N}$$



elements in between (0<j<N)

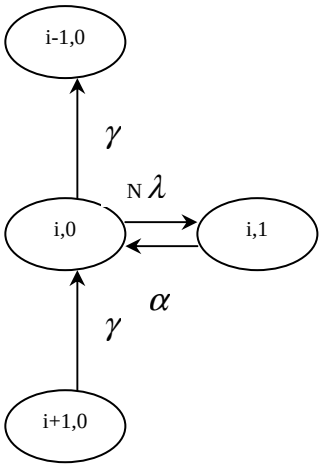
$$[j\beta + j\alpha + (N - j)\lambda]P_{0j} = \lambda[N - (j-1)]P_{0,j-1} + \alpha(j+1)P_{0,j+1} + \gamma P_{1j}$$
$$\Rightarrow P_{0j} = \lambda[N - (j-1)]P_{0,j-1} + [1 - j\beta - j\alpha - (N - j)\lambda]P_{0j} + [\alpha(j+1)]P_{0,j+1} + \gamma P_{1j}$$



If we have infinite buffer size, for the rows 2, 3 … we have three conditions:

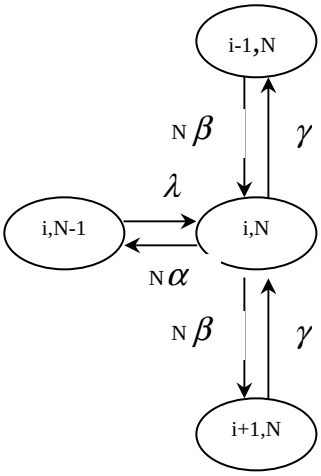
First element (j=0)

$$[N\lambda + \gamma]P_{i0} = \alpha P_{i1} + \gamma P_{i+1,0}$$
$$\Rightarrow P_{i,0} = (1 - N\lambda - \gamma)P_{i,0} + \alpha P_{i,1} + \gamma P_{i=1,0}$$



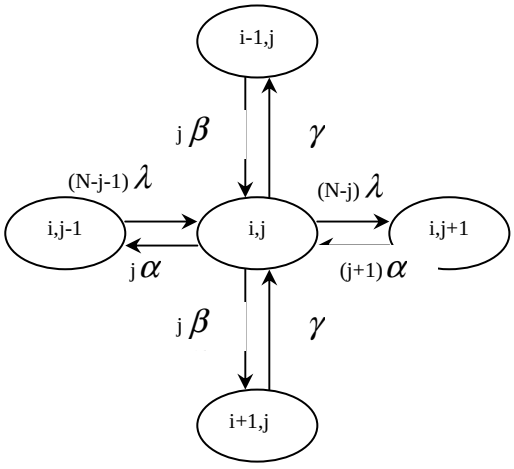
Last element (j=N)

$$P_{i,N}[N\alpha + N\beta + \gamma] = \lambda P_{i,N-1} + N\beta P_{i-1,N} + \gamma P_{i+1,N}$$
$$\Rightarrow P_{i,N} = \lambda P_{i,N-1} + P_{i,N}[1 - N\alpha - N\beta - \gamma] + N\beta P_{i-1,N} + \gamma P_{i+1,N}$$



Elements in between (0<j<N)

$$[\gamma + j\beta + j\alpha + (N - j)\lambda]P_{i,j} = \lambda[N - (j-1)]P_{i,j-1} + \alpha(j+1)P_{i,j+1} + j\beta P_{i-1,j} + \gamma P_{i+1,j}$$
$$\Rightarrow P_{i,j} = \lambda[N - (j-1)]P_{i,j-1} + [1 - \gamma - j\beta - j\alpha - (N - j)\lambda]P_{i,j} + \alpha(j+1)P_{i,j+1} + j\beta P_{i-1,j} + \gamma P_{i+1,j}$$



However, if our buffer size is finite we have the following three conditions for the last row:

First element (j=0)

$$[N\lambda + \gamma]P_{N,0} = \alpha P_{N,1}$$
$$=> P_{N,0} = (1 - N\lambda - \gamma)P_{N,0} + \alpha P_{N,1}$$

Last element (j=N)

$$P_{N,N}[N\alpha + \gamma] = \lambda P_{N,N-1} + N\beta P_{N-1,N}$$
$$=> P_{N,N} = \lambda P_{N,N-1} + P_{N,N}[1 - N\alpha - \gamma] + N\beta P_{N-1,N}$$

Elements in between (0<j<N)

$$[\gamma + j\alpha + (N - j)\lambda]P_{N,j} = \lambda[N - (j-1)]P_{N,j-1} + \alpha(j+1)P_{N,j+1} + j\beta P_{N-1,j}$$
$$=> P_{N,j} = \lambda[N - (j-1)]P_{N,j-1} + [1 - \gamma - j\alpha - (N - j)\lambda]P_{N,j} + \alpha(j+1)P_{N,j+1} + j\beta P_{N-1,j}$$

We will solve for infinite buffer since it is simpler.

The row vector can be defined as follows:

$$P_{i,\,j=0,1,\ldots N} = [P_{i,0} P_{i,1} \ldots\ldots P_{i,N}]$$

Where:

$$\boldsymbol{P_0 = P_0 B_0 + P_1 B_1}$$

$$B0 = \begin{bmatrix} 1-N\lambda & 0 & 0 & 0 & \ldots & 0 \\ \alpha & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & .. & .. & \ldots \\ 0 & 0 & \lambda(N-j+1) & 0 & \ldots & \ldots \\ 0 & 0 & 0 & 1-j\beta-j\alpha-\lambda(N-j) & 0 & \ldots & \ldots \\ \ldots & 0 & \alpha(j+1) & 0 & \ldots & \ldots \\ \ldots & \ldots & 0 & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \lambda \\ 0 & 0 & 0 & \ldots & 0 & 1-N\beta-N\alpha \end{bmatrix}$$

$$B1 = \begin{bmatrix} \gamma & 0 & \ldots & 0 \\ 0 & \gamma & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & \gamma \end{bmatrix}$$

$$P_i = P_{i-1} A_0 + P_i A_i + P_{i+1} A_2$$

Where:

$$A0 = \begin{bmatrix} 0 & \dots & \dots & \dots \\ 0 & \beta & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & j\beta & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & N\beta \end{bmatrix}$$

$$A1 = \begin{bmatrix} 1-N\lambda-\gamma & 0 & 0 & 0 & \dots & 0 \\ \alpha & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & .. & .. & \dots \\ 0 & 0 & \lambda(N-j+1) & 0 & \dots & \dots \\ 0 & 0 & 0 & 1-\gamma-j\beta-j\alpha-\lambda(N-j) & 0 & \dots & \dots \\ \dots & 0 & \alpha(j+1) & 0 & \dots & \dots \\ \dots & \dots & 0 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \lambda \\ 0 & 0 & 0 & \dots & 0 & 1-\lambda-N\beta-N\alpha \end{bmatrix}$$

$$A2 = \begin{bmatrix} \gamma & 0 & \dots & 0 \\ 0 & \gamma & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \gamma \end{bmatrix}$$

Using Neuts' general solution

$$R = A_0 + R A_1 + R^2 A_2$$

Where $\boldsymbol{R}$ is the minimum non-negative solution

We can simply find any row vector by using

$$P_0 = P_0 (B_0 + R B_1)$$

$$P_i = P_0 R^i$$

Now, the problem has been mapped to polynomial matrix equation that needs to be solved to find the matrix R. by minimum non-negative solution we mean that for any other non-negative solution R^, the matrix R is entry-wise less than the matrix R^.

The algorithmic method to solve for R

1. Initialization:

$$j = 0,\ A_0^{(0)} = A_0,\ A_1^{(0)} = A_1,\ A_2^{(0)} = A_2,\ A\hat{}^{(0)} = A_1$$

2. Compute the following:

$$A_0^{(j+1)} = -A_0^{(j)} (A_1^{(j)})^{-1} . A_0^{(j)}$$
$$A_1^{(j+1)} = A_1^{(j)} - A_0^{(j)} (A_1^{(j)})^{-1} A_2^{(j)} - A_2^{(j)} (A_1^{(j)})^{-1} A_0^{j}$$
$$A_2^{(j+1)} = -A_2^{(j)} (A_1^{(j)})^{-1} . A_2^{j}$$
$$A\hat{}^{(j+1)} = A\hat{}^{(j)} - A_2^{(j)} (A_1^{(j)})^{-1} A_0^{(j)},\ j \geq 0$$

3. If we have a matrix $M = (x_{i,j})_{i,j = 1, 2, \ldots, n}$, let us define

$$\|M\| = \max_{i=1, 2, \ldots, n} \sum_{j=1}^{n} 1 |b_{i,j}|$$

and we compute

$$r = \min\left\{\left\|A_0^{(j+1)}\right\|, \left\|A_2^{(j+1)}\right\|\right\}$$

4. If $r \geq \varepsilon$ then increment j and loop back to step 2, else

$$R\hat{} = -(A\hat{}^{(j+1)})^{-1} A_0$$

References

1. Neuts, M. F., “Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach,” *The John Hopkins University Press*, Baltimore, MD, 1981.

2. Foh, C.H. et al, “Modeling and Performance Evaluation of GPRS”.

3. He, C. Meini B., Rhee, N.L., “A shifted cyclic reduction algorithm for QBD”.

4. Meini, B., Fortran Software Package, http://fibonacci.dm.unipi.it/~meini/ric.html

5. Meini, B., Favati P., “Solving certain queueing problems by means of regular splittings,” *Appl. Math. Letters*, **13**:99-105, 2000

6. Meini, B., “Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method,” *Advances in Performance Analysis*, **1**:215-225, 1998.

7. Lipsky L., Schwefel, H.P., “N Burst processes as Models of Self-Similar traffic telecommunication systems – 1. Introduction and LAQT background.

8. Bini, D., Meini, B. and Latouche G., “Quadratically convergent algorithms for solving matrix polynomial equations”, to appear in *Linear Algebra Appl*.

9. Bini D. and Meini B., “Improved cyclic reduction for solving queueing problems.” *Numerical Algorithms*, **15**:57--74, 1997.

10. Meini, B. and Favati, P., “On functional iteration methods for solving nonlinear matrix equations arising in queueing problems,” *IMA Journal of Numerical Analysis*, 19:39-49, 1999.

Modeling and Performance Evaluation of GPRS

Chuan Heng Foh[1], Beatrice Meini[2], Bartek Wydrowski[1] and Moshe Zukerman[1]

[1]ARC Special Research Centre for Ultra-Broadband Information Networks,
EEE Department, The University of Melbourne,
Parkville, Vic. 3010, Australia
{chuanhf, b.wydrowski, m.zukerman}@ee.mu.oz.au

[2]Dipartimento di Matematica,
Universita' di Pisa, via Buonarroti 2,
56127 Pisa,
Italy
meini@dm.unipi.it

Abstract

This paper provides an accurate model of the General Packet Radio Service (GPRS). GPRS is modeled as a single server queue in a Markovian environment. The queueing performance of data packets is evaluated by matrix geometric methods. The arrival process is assumed to follow a two state Markov modulated Poisson process (MMPP), and the service rate fluctuates based on voice loading. The analytical results are confirmed by simulation.

1. Introduction

The rapid growth of the Internet has prompted a need for wireless data access to the Internet. Although Global System for Mobile Communications (GSM) systems provide a fixed rate data service, they result in inefficient use of bandwidth for data users due to the bursty nature of data traffic.

To carry data traffic more efficiently, the use of dynamic channel allocation for data packets was implemented in GSM, known as General Packet Radio Service (GPRS) [1]. Fundamentally, GPRS is based on the *hybrid switching* [2] principle with two types of traffic: (i) voice calls and (ii) packet data. In practice, in accordance with current design and traffic management policies, voice calls have priority over data packets and data packets which cannot immediately be transmitted are queued at the source.

In modeling GPRS, we assume for simplicity that all awaiting data packets in all remote sources are in a single server queue (SSQ). The bandwidth available to these data packets is dependent upon the number of voice calls in the system.

To capture the bursty nature of data traffic, we model the arrival process as a Markov Modulated Poisson Process (MMPP). It is shown that MMPP can be used to produce bursty traffic [3], and we further demonstrate the fundamentally significant short range dependent property of MMPP.

The single server queue is modeled as a queue in a Markovian environment to reflect both the MMPP arrival process and the effect of voice loading on the packet data queueing performance. We use matrix analytic methods [4] to obtain the numerical results of the delay of data packets. To verify our analytic model, we develop a simulation program that models the operation of GPRS with more details. It includes the signaling time required for data packets, and the non-exhaustive use of a time slot of a GSM TDMA frame.

This paper is structured as follows. In Section 2, GPRS is outlined. The analytic model is described in Section 3. In Section 4, the performance evaluation of GPRS is presented. Finally in Section 5, we verify the analytical results with the simulation results.

2. General Packet Radio Service

GPRS is a GSM packet radio service. GSM shares the radio spectrum resource by performing both frequency-division multiple access (FDMA) and time-division multiple access (TDMA). FDMA divides the 25Mhz spectrum into 124 carrier frequencies spaced 200khz apart. A certain number of these frequency bands is allocated to a base station of a cell. Each of these frequency bands is further divided in time. Eight channels are created by dividing time into eight time slots. A TDMA frame is formed by packing the eight time slots.

In the fixed rate data service of GSM, a data user is permanently allocated two time slots from every TDMA. One of the time slot is used for uplink and another for downlink. Due to the bursty nature of data packets, the fixed time slot allocation is inefficient.

GPRS employs the concept of capacity on demand by dynamically allocating a certain number of time slots to a data user whenever there are data packets waiting for transmissions. Furthermore, uplink and downlink are allocated separately so that more efficient bandwidth usage can be reached for asymmetric data traffic. For a cell that supports GPRS, all the radio resources are shared by both GSM voice users and GPRS data users. The unused time slots from voice traffic may be reallocated to carry GPRS traffic.

There are four coding options available for GPRS: CS-1, CS-2, CS-3 and CS-4 capable of transmitting 9.05k, 13.4k, 15.6k, and 21.4k bits in every second respectively. They are tailored to suit different channel conditions.

TABLE 1. Parameters of the MMPP traffic in Fig. 1

	(a)	(b)	(c)	(d)
λ_0	0.01	0.01	0.01	0.01
λ_1	1	0.1	0.05	0.05
r_0	10^{-8}	10^{-8}	10^{-5}	0.5
r_1	$2\cdot10^{-9}$	$2\cdot10^{-9}$	10^{-6}	0.005

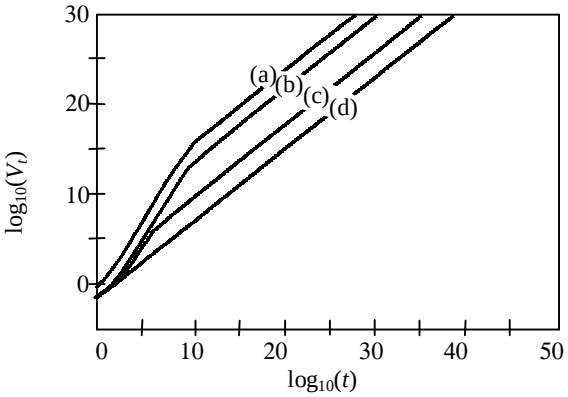


Fig. 1. The Variance Time Curve of MMPP

Readers are referred to [1] for more details of GPRS specifications.

3. Queueing Model

The packet data arrival traffic model

The arrival process is assumed to follow a two state MMPP [3]. A two state MMPP is an alternating Markovian process with two arrival states, where the arrival process in arrival state m is a Poisson process with rate: λ_m, m=0,1. The sojourn time in each arrival state is exponentially distributed with the mean sojourn time in arrival state 0 and 1 being r_0^{-1} and r_1^{-1} respectively. The values of the mean and the variance of MMPP traffic are given in [3]. Let N_t be the amount of work arriving during time interval (0, t). For MMPP, the mean of N_t, denoted E[N_t], is:

$$E[N_t] = \frac{\lambda_0 r_1 + \lambda_1 r_0}{r_0 + r_1} \cdot t$$

and the variance of N_t, denoted V_t, is:

$$V_t = E[N_t] \cdot \left(1 + \frac{2(\lambda_0 - \lambda_1)^2 r_0 r_1}{(r_0 + r_1)^2 (\lambda_0 r_1 + \lambda_1 r_0)} - \frac{2(\lambda_0 - \lambda_1)^2 r_0 r_1}{(r_0 + r_1)^3 (\lambda_0 r_1 + \lambda_1 r_0) \cdot t} \left(1 - e^{-(r_0 + r_1)t}\right) \right)$$

It is well known that the autocorrelation of the arrival process has significant effect on queueing performance. It is also well known that real traffic (data and VBR video) exhibits long range dependence (LRD) [5]. However, since buffer size is limited so is the time period over which autocorrelation has effect. The larger the buffer size, the longer is the time period over which autocorrelation affects queueing performance. If the buffer size is equal to zero, autocorrelations has no effect on performance.

If we accept the view that for a given buffer size, the shape of autocorrelation curve, from a certain point onwards, does not affect queueing performance, we can use the MMPP, which is a short range dependent (SRD) process, to model LRD real traffic for the purpose of queueing performance evaluation.

For that purpose, we will consider the variance time curve of MMPP. In Fig. 1, we plot four variance time curves with different parameter sets of MMPP traffic. By comparing the four presented curves, the critical time interval of the traffic and the slopes of the curve within the critical interval can be controlled by the parameters of MMPP. For curves (a), (b) and (c), we obtained SRD traffic with different critical time intervals.

The voice traffic model

A Voice call is allocated a channel for the duration of the call. Regardless of whether there is silence or activity a slot is dedicated to the voice call for the duration of the call. Therefore admission of a voice call decreases the number of channels (servers) available for the data packets and the departure of a call increases the number of available channels (servers) available for the data packets.

The departure and arrival process of the voice traffic is modeled as an M/M/k/k process, where k is the number of voice channels present in the system. The voice arrival process is Poisson with parameter λ_v and the voice call holding time is assumed exponential with mean $1/\mu_v$.

4. Queueing Analysis

We have modeled GPRS as a queue in a Markovian environment, and we follow Neuts' analysis of such a queueing model as described in [4] pages 254-264. The infinitesimal generator is first introduced here to describe the system, then, we review Neuts' solution as applied to our queueing problem, and finally we show how the rate matrix R, required for Neuts' solution, is computed.

Infinitesimal Generator

We will assume that there are c channels (c does not include the channels allocated for signaling), out of total c channels we assign d channels exclusively for data packets. Therefore only c-d channels are available for circuit switched voice traffic. The state of the system under consideration is denoted by the three dimensional vector (i,j,m) where i is the number of data packets in the queue (including the one in service), j is the number of channels available for data packets, m is the arrival state (m takes the values 0 and 1). The service rate is always equal to $j\mu$, where μ is the service rate provided by one

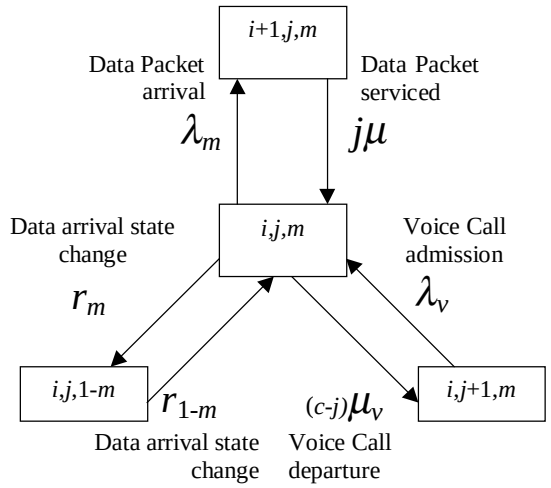


Fig. 2. State Transition Diagram

data channel, and $j = d, d+1, d+2, \ldots, c$. The packet data arrival rate is λ_m, $m = 0, 1$. All the possible state transitions are presented in Fig. 2. Hence, Fig. 2 defines the infinitesimal generator matrix G.

The sum of the entries in each row of G is 0. Let $\hat{h}_{i,j,m}$ be the probability of being in state (i, j, m), then the vector $\hat{h}$ is:

$$(\hat{h}_{0,0,0}, \hat{h}_{0,0,1}, \hat{h}_{0,1,0}, \hat{h}_{0,1,1}, \ldots,$$
$$\hat{h}_{0,c,1}, \hat{h}_{1,0,0}, \hat{h}_{1,0,1}, \hat{h}_{1,1,0}, \hat{h}_{1,1,1}, \ldots,$$
$$\hat{h}_{1,c,1}, \hat{h}_{2,0,0}, \hat{h}_{2,0,1}, \hat{h}_{2,1,0}, \hat{h}_{2,1,1}, \ldots).$$

The state transition balance equation is then $\hat{h}G = 0$. The steady state queue size distribution x_i is related to $\hat{h}$ as such $x_i = \sum_m \sum_j \hat{h}_{i,j,m}$. We now obtain x_i by another approach, using Neuts' analysis.

Neuts' solution

The steady state queue size distribution vector $\boldsymbol{x}$ can be solved analytically by considering the data packet SSQ's parameters being driven by a Markovian environment. The Markovian environment is comprised of two processes, the voice call arrivals and departures as well as the transition between data arrival states 0 and 1. These processes are independent, and determine the state of the environment j and m. The transition probability matrix of the Markovian environment is denoted Q and it is shown in Fig. 3.

The Markovian random environment is the stochastic process that determines the number of voice calls in the system and the packet data arrival state. As discussed, the packet data service rate fluctuates based on the voice calls in the system. For each state of the environment, there is an appropriate data service rate in vector $\boldsymbol{\mu}$ and packet data arrival rate in vector $\boldsymbol{\lambda}$.

The packet data service rates in each state of the environment (j, m) are:

$\boldsymbol{\mu}_{j,m} = \boldsymbol{\mu}_j = j\mu$, m=0,1 and $j=d, d+1, d+2, ..., c$.

Let vector $\boldsymbol{\mu}$ be defined by

$\boldsymbol{\mu} = (d\mu, (d+1)\mu, (d+2)\mu, \ldots, c\mu)$.

The packet data arrival rates in each state of the environment (j, m) are:

$\boldsymbol{\lambda}_{j,m} = \lambda_m$, m=0,1 and $j=d, d+1, d+2, ..., c$.

Let vector $\boldsymbol{\lambda}$ be defined by

$\boldsymbol{\lambda} = (\lambda_0, \lambda_1, \lambda_0, \lambda_1, \lambda_0, \lambda_1, \ldots. \lambda_1)$.

For example, if j=10 and m=1, with c=22 and d=1, there are 22 channels with 1 reserved for data packets, and there are 12 voice calls in the system as there are 10 packet data channels available. The packet data arrival state is 1. The data service process then has parameter 10μ (since 10 channels are allocated to serving data) and the data arrival process has parameter λ_1.

Assuming the queue is stable (mean arrival rate < mean service rate), to determine the queueing performance of the GPRS system, the stationary probability vector $\boldsymbol{x}=(x_0,x_1,x_2,\ldots)$, which describes the probability distribution of the queue size, needs to be determined.

Using Neuts' formalisation of the M/M/1 queue in a Markovian environment [4], this problem is translated into finding the minimal solution for the matrix R, which satisfies the matrix equation

$$R^2\Delta(\mu) + R(Q - \Delta(\lambda + \mu)) + \Delta(\lambda) = 0$$

where Q is the transition probability matrix of the Markovian environment, and $\Delta(z)$ the diagonal matrix of the vector $\boldsymbol{z}$.

The matrix R can be evaluated using a cyclic reduction

	d,0	d,1	d+1,0	d+1,1	…	c,1
d,0	$-r_0-(c-d)\mu_v$	r_0	$(c-d)\mu_v$			
d,1	r_1	$-r_1-(c-d)\mu_v$		$(c-d)\mu_v$		
d+1,0	λ_v		$-\lambda_v-r_0-(c-d-1)\mu_v$	r_0		
d+1,1		λ_v	r_1	$-\lambda_v-r_1-(c-d-1)\mu_v$		
…						
c,1						$-\lambda_v-r_1$

Fig. 3. The Matrix Q

algorithm described in the next subsection. The stationary probability vector $\boldsymbol{x}$ of the stable queue is given by

$$x_k = \pi(I-R)R^k \quad for\ k \geq 0$$

where π is the stationary probability vector of Q (Eq. (6.2.5) from [4]). The vector π is given by solving $\pi \cdot Q = 0$ by, for example, successive relaxation.

The mean queue size is computed using the stationary probability vector $\boldsymbol{x}$. The mean data packet delay is found using Little's law. Packet delay is the time from the generation of the packet to the time the last bit is sent. The mean delay is thus obtained as follows:

$$mean\ packet\ arrival\ rate = \frac{\lambda_0 r_1 + \lambda_1 r_0}{r_0 + r_1}$$

$$mean\ queue\ size = \sum_{i=1}^{i=\infty} x_i \cdot i$$

$$mean\ delay = \frac{mean\ queue\ size}{mean\ packet\ arrival\ rate}.$$

Computation of the rate matrix R

In this subsection we describe the algorithm for the computation of the rate matrix R. The matrix R is the minimal nonnegative solution of the matrix equation

$$A_0 + A_1 R + A_2 R^2 = 0$$

where $A_0 = \Delta(\mu)$, $A_1 = Q - \Delta(\lambda+\mu)$, $A_2 = \Delta(\lambda)$. Here, minimal nonnegative solution means that for any other nonnegative solution $\hat{R}$, the matrix R is entrywise less than the matrix $\hat{R}$.

This effective numerical method that works in the case where the associated M/M/1 queue is transient or positive recurrent [4]. The method is based on the use of cyclic reduction and is extensively discussed in the early paper [7] and later in [8, 6, 9]. Here we will recall the algorithm and its convergence properties and we refer the reader to [7, 6] for details and proofs.

This method has nice properties, namely it is quadratically convergent, that is the approximation error e_j at the j-th step is such that $e_j \leq c\sigma^{2^j}$ for suitable constants $c>0$ and $0<\sigma<1$, and has a very low computational cost. Moreover all the computations involved in this scheme are numerically stable, since they consist of sums and products of nonnegative matrices and inversions of M-matrices (see [7]).

The algorithm consists of generating the four sequences of matrices $\{A_0^{(j)}\}_{j\geq 0}$, $\{A_1^{(j)}\}_{j\geq 0}$, $\{A_2^{(j)}\}_{j\geq 0}$, $\{\hat{A}^{(j)}\}_{j\geq 0}$ according to the following recurrences:

$$\begin{aligned}
A_0^{(j+1)} &= -A_0^{(j)}\left(A_1^{(j)}\right)^{-1}A_0^{(j)}\\
A_1^{(j+1)} &= A_1^{(j)} - A_0^{(j)}\left(A_1^{(j)}\right)^{-1}A_2^{(j)} - A_2^{(j)}\left(A_1^{(j)}\right)^{-1}A_0^{(j)}\\
A_2^{(j+1)} &= -A_2^{(j)}\left(A_1^{(j)}\right)^{-1}A_2^{(j)}\\
\hat{A}^{(j+1)} &= \hat{A}^{(j)} - A_2^{(j)}\left(A_1^{(j)}\right)^{-1}A_0^{(j)},\ j \geq 0,\\
A_0^{(0)} &= A_0,\ A_1^{(0)} = A_1,\ A_2^{(0)} = A_2,\ \hat{A}^{(0)} = A_1.
\end{aligned} \tag{1}$$

In [2, 1] it is shown that these sequences of matrices are such that:

1. for any $j\geq 0$ it holds
$$\hat{A}^{(j)}R + A_2^{(j)}R^{2^j+1} = -A_0;$$
2. if the associated M/M/1 queue is positive recurrent, then the sequences $\{A_0^{(j)}\}_{j\geq 0}$ converges in norm to zero as σ^{2^j}, where $\sigma = \max\{|\alpha|:\det(A_0+\alpha A_1+\alpha^2 A_2) = 0,\ \alpha\in C,\ |\alpha|<1\}$ is the largest modulus eigenvalue of R;
3. if the associated M/M/1 queue is transient, then the sequence $\{A_2^{(j)}\}_{j\geq 0}$ converges in norm to zero as σ^{2^j}, where $\sigma = 1/\min\{|\alpha|:\det(A_2+\alpha A_1+\alpha^2 A_0) = 0,\ \alpha\in C,\ |\alpha|>1\}$;
4. the sequence $\left\{\left(\hat{A}^{(j)}\right)^{-1}A_2^{(j)}R^{2^j+1}\right\}_{j\geq 0}$ converges in norm to zero as σ^{2^j}, where $0<\sigma<1$ is one of the two values defined above.

From these properties, an approximation of R is given by $-\left(\hat{A}^{(j)}\right)^{-1}A_0$, for a moderately large value of j. Moreover, in order to stop the computation, we check the minimum value between the norms of $A_0^{(j)}$ and $A_2^{(j)}$: if this value is smaller than a threshold value ε, then $-\left(\hat{A}^{(j)}\right)^{-1}A_0$ will be an approximation of R.

The resulting algorithm can be synthesized by the following scheme:

Algorithm: Cyclic reduction for computing *R*

Input The matrices A_0, A_1, A_2 and an error bound $\varepsilon > 0$ for the stopping condition.

Output An approximation $\tilde{R}$ of R.

Computation

1. Set $j = 0$, $A_0^{(0)} = A_0$, $A_1^{(0)} = A_1$, $A_2^{(0)} = A_2$, $\hat{A}^{(0)} = A_1$.
2. Compute $A_0^{(j+1)}$, $A_1^{(j+1)}$, $A_2^{(j+1)}$, $\hat{A}^{(j+1)}$ by means of (1) and $r = \min\left\{\left\|A_0^{(j+1)}\right\|, \left\|A_2^{(j+1)}\right\|\right\}$, where, for a matrix $B = (b_{h,k})_{h,k=1,\ldots,n}$, $\|B\| = \max_{h=1,\ldots,n}\sum_{k=1}^{n}\left|b_{h,k}\right|$.
3. If $r \geq \varepsilon$ set $j = j + 1$ and go back to step 2; otherwise, set $\tilde{R} = -\left(\hat{A}^{(j+1)}\right)^{-1}A_0$.

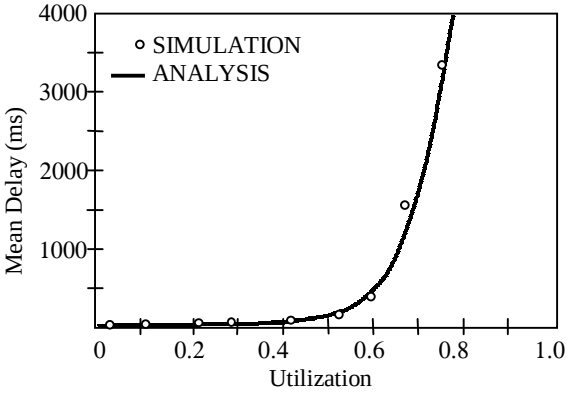


Fig. 4. Mean delay versus utilization

5. Simulation Testing

The analysis was performed for a typical cell. In a typical cell, there are 24 available channels, two of which are often reserved for signaling and Short Message Service (SMS). Of the 22 available, one is reserved exclusively for packet data traffic, leaving 21 for voice traffic. The mean packet length was set to 512 bytes, a typical TCP packet, making the mean data service rate μ=512*8/45.25=1/90.519 packets/burst period. The mean voice call holding time was taken to be 36000 burst periods long (3 minutes), and the call arrival rate was chosen so that the cell is 30% utilized by voice calls λ_v=0.3·(c-d)·μ_v. Table 2 lists all of the parameters used. These parameters were used to obtain a result using the analytical approach as well as by simulation.

TABLE 2. Model Parameters

Parameter	Value	Meaning
λ_0	variable	Packet arrival rate state 0
λ_1	$\lambda_1=5\lambda_0$	Packet arrival rate state 1
r_0	0.00001	State 0 to 1 transition rate
r_1	0.000001	State 1 to 0 transition rate
μ	1/90.519	Mean service rate
c	22	Total channels
d	1	Data reserved channels
μ_v	1/36000	Voice call 1/(hold time)
λ_v	0.3·(c-d)·μ_v	Voice call arrival rate

The simulation includes signaling delays not modeled in the analysis. To account for resource allocation, each data packet generated suffers a fixed delay of 1 GSM TDMA frame. If a transmission (data or voice) ends during a time slot, the remaining time of this time slot cannot be used for further voice or data traffic.

Packet delay results obtained using the parameters in Table 2 are shown in Fig. 4 for both the analysis and simulation. The agreement between the analytical model and simulation validates the analytical approach.

Interestingly, it appears that the additional overheads that the simulation accounted for had an insignificant impact on the mean queueing delay. Resource allocation for a packet transmission is suffered only by the first TDMA frame of the packet. For packets spanning many frames, this results in insignificant additional delay. The non exhaustive use of time slots also did not contribute to any significant delays. This is also true as long as voice or packet transmissions last over many time slots.

Fig. 4 indicates that in order to operate GPRS with reasonable packet delays the utilization must be kept below 60%.

6. Conclusion

The focus of this work was to present a simple and accurate analytical model of the GPRS system. The use of MMPP to obtain SRD traffic, captured the bursty nature of packet data traffic. Matrix geometric techniques were used to obtain queueing performance results. Simulation testing agreed with the analytical model results.

References

[1] R. Kalden, I. Meirick and M. Meyer, "Wireless Access Based on GPRS," *IEEE Personal Communications*, April 2000.

[2] M. Zukerman, "Circuit allocation and overload control in a hybrid switching system," *Computer Networks and ISDN Systems*, vol. 16, no. 4, 1989, pp. 281-298.

[3] H. Heffes and D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE JSAC*, vol. SAC-4, no. 6, September 1986.

[4] M. F. Neuts, "Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach," *The Johns Hopkins University Press*, Baltimore, MD, 1981.

[5] R. G. Addie, M. Zukerman and T. D. Neame, "Broadband traffic modeling: simple solutions to hard problems," *IEEE Communication Magazine*, August 1998, pp. 88-95.

[6] D. A. Bini, L. Gemignani, and B. Meini, "Computations with infinite Toeplitz matrices and polynomials," *Linear Algebra Appl.*, 2001.

[7] D. A. Bini and B. Meini, "On the solution of a nonlinear matrix equation arising in queueing problems," *SIAM J. Matrix Anal. Appl.*, 17:906-926, 1996.

[8] D. A. Bini and B. Meini, "Improved cyclic reduction for solving queueing problems," *Numerical Algorithms*, 15:57-74, 1997.

[9] C. He, B. Meini, and N. H. Rhee. "A shifted cyclic reduction algorithm for QBD's," *SIAM J. Matrix Anal. Appl.*, 2001.

A shifted cyclic reduction algorithm for QBDs

C. He* B. Meini† N.H. Rhee‡

May 9, 2000

Abstract

The problem of the computation of the rate matrix G associated with discrete-time QBD Markov chains is analyzed. We present a shifted cyclic reduction algorithm and show that the speed of convergence of the latter modified algorithm is always faster than that of the original cyclic reduction.

1 Introduction

The block tridiagonal and block Toeplitz structure of the probability transition matrix P associated with QBD problems, i.e.,

$$P = \begin{bmatrix} B_0 & B_1 & & 0 \\ A_0 & A_1 & A_2 & \\ & A_0 & A_1 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix},$$

allowed the design of fast and reliable methods for computation of the rate matrix G [16, 11, 2, 3, 4, 12, 5]. Here, A_0, A_1, A_2, B_0, B_1 are $k \times k$ nonnegative matrices such that $A_0 + A_1 + A_2$ is irreducible, and P is irreducible, stochastic and positive recurrent. These fast methods rely on the property that the matrix G, which solves the nonlinear matrix equation

$$G = A_0 + A_1 G + A_2 G^2, \tag{1}$$

*Sprint Corporation, Network Planning and Design, 7171 W. 95th Street Overland Park, KS 66212, charlie.he@mail.sprint.com

†Dipartimento di Matematica, Università di Pisa, Italy, meini@dm.unipi.it

‡Department of Mathematics and Statistics, University of Missouri at Kansas City, Kansas City, MO 64110, rhee@cctr.umkc.edu

can be computed by solving the infinite block tridiagonal, block Toeplitz system

$$\begin{bmatrix} I-A_1 & -A_2 & & 0 \\ -A_0 & I-A_1 & -A_2 & \\ & -A_0 & I-A_1 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^2 \\ G^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}. \tag{2}$$

Recently, in [2, 3, 4], Bini and Meini have devised a new quadratically convergent and numerically stable algorithm for the computation of G, based on a functional representation of cyclic reduction, which applies to general M/G/1 type Markov chains [16] and which extends the method of Latouche and Ramaswami [11].

The aim of this paper is to introduce a shifted cyclic reduction algorithm for QBDs and to show that, under the nonrestrictive assumption that G has only one eigenvalue of modulus one, the speed of convergence of a shifted cyclic reduction algorithm is always faster than that of the original cyclic reduction algorithm.

More precisely, since G has a known eigenvalue (equal to one) and a known corresponding right eigenvector, we may apply a shifting technique, that consists in moving the eigenvalue 1 to 0, and in maintaining the remaining ones. This trick leads to a new quadratic matrix equation, having as solution a singular matrix H, such that $G = H + \boldsymbol{e}\boldsymbol{u}^T$, where $\boldsymbol{e}$ is the vector having all the entries equal to 1, and $\boldsymbol{u}$ is a known arbitrary vector, with positive entries, such that $\boldsymbol{u}^T\boldsymbol{e} = 1$. Thus the problem of the computation of G is reduced to the problem of the computation of H, that shares with G all the eigenvalues, except for the eigenvalue 1, that is moved to 0.

In order to compute H we apply the cyclic reduction algorithm, and we show that the shifting leads to a faster convergence. More precisely, in standard cyclic reduction, the error of the approximation to G, for $j \to \infty$, goes to zero as $O\left((1/\sigma)^{2^j}\right)$, where $\sigma = \min\{|\lambda| : \ |\lambda| > 1, \ \phi(\lambda) = 0\}$, and $\phi(\lambda) = \det(-A_0 + (I - A_1)\lambda - A_2\lambda^2)$. With the shifting technique, the error goes to zero as $O\left((\theta/\sigma)^{2^j}\right)$, where θ is any real number such that $\max\{|\lambda| : \ |\lambda| < 1, \ \phi(\lambda) = 0\} < \theta < 1$. Thus the convergence speed can be much increased; in particular the improvement is more appreciated if $\sigma \approx 1$ and $\theta << 1$.

Finally, we introduce a means to measure the conditioning of the quadratic matrix equations, and we prove that the shifted equation is better conditioned than the original one. Indeed, the shifting technique leads to a better

rate of convergence, but destroys the nonnegativity and the M-matrix properties of the blocks generated at each step of standard cyclic reduction, and in principle this fact could lead to a loss of accuracy of the results.

We have performed several numerical experiments, which show that the shifted cyclic reduction algorithm is fast and numerically accurate.

The paper is organized as follows. In Section 2 we recall the cyclic reduction algorithm for QBDs. In Section 3 we apply the shifting technique, and we show how the solutions of the shifted matrix equation are related to the solution of the original one. In Section 4 we analyze the convergence properties of the cyclic reduction algorithm applied to the shifted matrix equation. In Section 5 we study the conditioning of the two matrix equations. In Section 6 we present some numerical results.

2 The cyclic reduction algorithm

In this section we recall the cyclic reduction algorithm for QBDs, described in [3].

Let us consider the system (2). By recursively applying block cyclic reduction, i.e., an odd-even permutation of block rows and block columns, followed by one step of Gaussian elimination, the following sequence of infinite block tridiagonal systems is generated:

$$\begin{bmatrix} I-\widehat{A}_1^{(j)} & -A_2^{(j)} & & 0 \\ -A_0^{(j)} & I-A_1^{(j)} & -A_2^{(j)} & \\ & -A_0^{(j)} & I-A_1^{(j)} & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^{2^j+1} \\ G^{2\cdot 2^j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad j \geq 0, \tag{3}$$

where $A_0^{(0)} = A_0$, $\widehat{A}_1^{(0)} = A_1^{(0)} = A_1$, $A_2^{(0)} = A_2$, and the blocks $\widehat{A}_1^{(j)}$, $A_i^{(j)}$, $i = 0, 1, 2$, $j \geq 1$, are defined by the recurrences:

$$\begin{aligned} A_0^{(j+1)} &= A_0^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)} \\ A_1^{(j+1)} &= A_1^{(j)} + A_0^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_2^{(j)} + A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)} \\ A_2^{(j+1)} &= A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_2^{(j)} \\ \widehat{A}_1^{(j+1)} &= \widehat{A}_1^{(j)} + A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)}. \end{aligned} \tag{4}$$

The sequences of matrices generated by the above relations allow the fast computation of the matrix G. Indeed, from (3) it follows that

$$(I - \widehat{A}_1^{(j)})G - A_2^{(j)} G^{2^j+1} = A_0. \tag{5}$$

On the other hand, if we denote by R the minimal nonnegative solution of the matrix equation $R = A_2 + RA_1 + R^2A_0$, then the following equation is verified (see [3, 11])

$$-A_2^{(j)} + R^{2^j}(I - A_1^{(j)}) - R^{2\cdot 2^j}A_0^{(j)} = 0.$$

Moreover, the maximum modulus eigenvalue of R is real, simple and unique, and it is equal to $1/\sigma$ (see [8]), where $\sigma = \min\{|\lambda| : \ |\lambda| > 1, \ \phi(\lambda) = 0\}$, and $\phi(\lambda) = \det(-A_0 + (I - A_1)\lambda - A_2\lambda^2)$. Thus, there exists an operator norm $||\cdot||$ such that $||R|| = \rho(R)$, where $\rho(R)$ denotes the spectral radius of R; hence, we obtain

$$||A_2^{(j)}|| \leq \rho(R)^{2^j}(||I - A_1^{(j)}|| + \rho(R)^{2^j}||A_0^{(j)}||).$$

Since the matrices $A_1^{(j)}$ and $A_0^{(j)}$ are bounded in norm [3], it follows that $||A_2^{(j)}|| = O\left(\rho(R)^{2^j}\right)$ and from (5), since G^{2^j+1} is bounded, that

$$(I - \widehat{A}_1^{(j)})G - A_0 = O\left(\left(\frac{1}{\sigma}\right)^{2^j}\right). \tag{6}$$

Hence an approximation of the matrix G is given by $\left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$, for a sufficiently large value of j.

Due to the double exponential convergence to zero of the sequence $A_2^{(j)}$, just a small number of steps can be sufficient to reach a good approximation of G.

The rate of convergence is given by $1/\sigma$, and it is therefore related to the closeness to the unit circle of the smallest zero of $\phi(\lambda)$ of modulus larger than one.

In the next section we present a trick to improve the rate of convergence; indeed, we show that by applying a deflating technique, that consists in removing the zero $\lambda = 1$ of $\phi(\lambda)$, the rate of convergence can be reduced to θ/σ, where θ is any real number such that $\max\{|\lambda| : \ |\lambda| < 1, \ \phi(\lambda) = 0\} < \theta < 1$.

3 A shifted matrix equation

In the sequel we assume, without loss of generality, that $\lambda = 1$ is the only zero of the function $\det(-A_0 + (I - A_1)\lambda - A_2\lambda^2) = 0$ on the unit circle [7].

Since G has a known eigenvalue 1 and a known eigenvector $\boldsymbol{e}$ where $\boldsymbol{e}$ is the k-dimensional vector having all the entries equal to 1, we can modify

G so that the eigenvalue 1 is shifted to 0. This shifting technique improves the convergence speed of the cyclic reduction algorithm.

Set $H = G - \boldsymbol{e}\boldsymbol{u}^T$, where $\boldsymbol{u}$ is any vector whose elements are positive and such that $\boldsymbol{u}^T\boldsymbol{e} = 1$. Then the eigenvalues of H are those of G except that in H the eigenvalue 1 of G is replaced by 0. Moreover, $\boldsymbol{e}$ is an eigenvector of H corresponding to the eigenvalue 0, and hence $H\boldsymbol{e} = 0$. So we have

$$G = H + \boldsymbol{e}\boldsymbol{u}^T \quad \text{and} \quad G^2 = (H + \boldsymbol{e}\boldsymbol{u}^T)^2 = H^2 + \boldsymbol{e}\boldsymbol{u}^T H + \boldsymbol{e}\boldsymbol{u}^T.$$

By replacing G by H in (1) we obtain that H solves the following shifted equation

$$B_0 + B_1 H + B_2 H^2 = H, \tag{7}$$

where

$$\begin{aligned} B_0 &= A_0 + (A_1 + A_2 - I)\boldsymbol{e}\boldsymbol{u}^T = A_0(I - \boldsymbol{e}\boldsymbol{u}^T) \\ B_1 &= A_1 + A_2\boldsymbol{e}\boldsymbol{u}^T \\ B_2 &= A_2. \end{aligned} \tag{8}$$

So the computation of the matrix G can be reduced to the computation of the matrix $H = G - \boldsymbol{e}\boldsymbol{u}^T$, that solves the nonlinear matrix equation (7).

3.1 Spectral properties of the shifted matrix polynomial

The rank one corrections of the matrices A_0 and A_1 have the effect to keep unchanged all the zeros of the polynomial $\phi(\lambda) = \det A(\lambda)$, $A(\lambda) = -A_0 + (I - A_1)\lambda - A_2\lambda^2$, except for the zero $\lambda = 1$, that is moved to $\lambda = 0$. Indeed, the following result holds:

Theorem 1 *The zeros of the polynomial $\psi(\lambda) = \det B(\lambda)$, $B(\lambda) = -B_0 + (I - B_1)\lambda - B_2\lambda^2$, are*

$$\{\lambda \ : \ \det A(\lambda) = 0, \ \lambda \neq 1\} \cup \{0\}.$$

To prove the above theorem we need to introduce some notations and some results on pairs of matrices (see, for example, [9]).

Definition 1 *Let A and B be $n \times n$ complex matrices. We define $\Lambda(A, B)$ by*

$$\Lambda(A, B) = \{\lambda \in \mathbf{C} : \det(A - \lambda B) = 0\}.$$

If $\lambda \in \Lambda(A, B)$, λ is called an eigenvalue of the pair (A, B). If λ is an eigenvalue of the pair (A, B), then there exists a nonzero vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \lambda B\boldsymbol{x}$, and such an $\boldsymbol{x}$ is called an eigenvector *of the pair (A, B) corresponding to the eigenvalue λ.*

Theorem 2 *If A and B are $n \times n$ complex matrices, then there exist unitary matrices U and V such that $U^H A V = R$ and $U^H B V = \tilde{R}$ where R and $\tilde{R}$ are upper triangular. If for some i, r_{ii} and $\tilde{r}_{ii}$ are both zero, then $\Lambda(A,B) = \mathbf{C}$. Otherwise*

$$\Lambda(A,B) = \{r_{ii}/\tilde{r}_{ii} : \tilde{r}_{ii} \neq 0,\ i = 1, \ldots, n\}.$$

Note that $\boldsymbol{v}_1$, the first column of V, is an eigenvector of the pair (A,B) corresponding to the eigenvalue $r_{11}/\tilde{r}_{11}$.

Definition 2 *Two pairs of matrices (A,B) and (C,D) are said to be* equivalent *if there exist nonsingular matrices L and M such that $C = LAM$ and $D = LBM$. It is easy to verify that if (A,B) and (C,D) are equivalent pairs, then $\Lambda(A,B) = \Lambda(C,D)$.*

Let us now prove Theorem 1:
Proof of Theorem 1. Let

$$\Lambda(A) = \{\lambda \in \mathbf{C} : \det A(\lambda) = 0\}, \quad \Lambda(B) = \{\lambda \in \mathbf{C} : \det B(\lambda) = 0\}.$$

If we set

$$S = \begin{bmatrix} I - A_1 & -A_0 \\ I & 0 \end{bmatrix} \qquad \text{and} \qquad T = \begin{bmatrix} A_2 & 0 \\ 0 & I \end{bmatrix}$$

it is easy to verify that $\Lambda(A) = \Lambda(S,T)$. Similarly, if we set

$$\tilde{S} = \begin{bmatrix} I - B_1 & -B_0 \\ I & 0 \end{bmatrix}$$

then $\Lambda(B) = \Lambda(\tilde{S},T)$.

Now if we let

$$L = \begin{bmatrix} I & -A_2 \boldsymbol{e}\boldsymbol{u}^T \\ 0 & I \end{bmatrix} \qquad \text{and} \qquad M = \begin{bmatrix} I & \boldsymbol{e}\boldsymbol{u}^T \\ 0 & I \end{bmatrix}$$

then by direct computation one can verify that

$$\tilde{S} = L(S - T \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix} [\mathbf{0}, \boldsymbol{u}^T])M \qquad \text{and} \qquad T = LTM.$$

So

$$\Lambda(B) = \Lambda(\tilde{S},T) = \Lambda(S - T \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix} [\mathbf{0}, \boldsymbol{u}^T], T).$$

Note that the vector $[\boldsymbol{e}^T, \boldsymbol{e}^T]^T$ is an eigenvector of the pair (S,T) corresponding to the eigenvalue 1. So there exist unitary matrices U and V such that

$$U^H S V = R \qquad \text{and} \qquad U^H T V = \tilde{R}$$

with

$$\boldsymbol{v}_1 = \frac{1}{\sqrt{2k}} \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix}.$$

Since $\boldsymbol{v}_1$ is an eigenvector of the pair (S,T) corresponding to the eigenvalue 1, we have $r_{11}/\tilde{r}_{11} = 1$. By Theorem 2 we know that

$$\begin{aligned} \Lambda(A) &= \Lambda(S,T) \\ &= \{\tfrac{r_{11}}{\tilde{r}_{11}}, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\} \\ &= \{1, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\}. \end{aligned} \tag{9}$$

Here we assumed that $\tilde{r}_{i,i} = 0$ for $i = l+1, \ldots, 2k$.

Since $U^H T [\boldsymbol{e}^T, \boldsymbol{e}^T]^T = \tilde{R} V^H (\sqrt{2k}\boldsymbol{v}_1) = \sqrt{2k}\tilde{r}_{11}\boldsymbol{e}_1$, where $\boldsymbol{e}_1$ is the first standard unit vector of length $2k$, and the first component of the row vector $[\boldsymbol{0}, \boldsymbol{u}^T]V$ is $1/\sqrt{2k}$, it follows that the matrix

$$U^H \left\{ S - T \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix} [\boldsymbol{0}, \boldsymbol{u}^T] \right\} V = R - U^H T \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix} [\boldsymbol{0}, \boldsymbol{u}^T] V$$

is upper triangular and has $r_{11} - \tilde{r}_{11}$, r_{22}, ..., $r_{2k,2k}$ as its diagonal elements. Hence by Theorem 2 we have

$$\begin{aligned} \Lambda(B) &= \Lambda(\tilde{S}, T) \\ &= \Lambda(S - T \begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{e} \end{bmatrix} [\boldsymbol{0}, \boldsymbol{u}^T], T) \\ &= \{\tfrac{r_{11} - \tilde{r}_{11}}{\tilde{r}_{11}}, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\} \\ &= \{0, \tfrac{r_{22}}{\tilde{r}_{22}}, \ldots, \tfrac{r_{l,l}}{\tilde{r}_{l,l}}\}. \end{aligned} \tag{10}$$

From (9) and (10) we see that $\Lambda(B)$ is the same as $\Lambda(A)$ except that in $\Lambda(B)$ $1 \in \Lambda(A)$ is replaced by 0. □

3.2 Solutions of the shifted matrix equation

The blocks B_0, B_1, B_2 are such that $H = G - \boldsymbol{e}\boldsymbol{u}^T$ solves the nonlinear matrix equation (7). The following theorem shows how the minimal nonnegative solutions of the nonlinear matrix equations (11) are transformed by the shifting:

Theorem 3 *Let $\boldsymbol{u}$ be positive, and let G, R, S and F be the minimal non-negative solutions of the matrix equations*

$$\begin{aligned} G &= A_0 + A_1 G + A_2 G^2 \\ S &= A_0 S^2 + A_1 S + A_2 \\ R &= R^2 A_0 + R A_1 + A_2 \\ F &= A_0 + F A_1 + F^2 A_2, \end{aligned} \tag{11}$$

respectively. Then $H = G - \boldsymbol{e}\boldsymbol{u}^T$, $K = (I - \boldsymbol{e}\boldsymbol{u}^T S)S(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1}$, $T = R$ and $V = F - \boldsymbol{w}\boldsymbol{u}^T(I - A_1 - A_0 S)^{-1}$, where $\boldsymbol{w} = (1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1} A_0 (I - S)\boldsymbol{e}$, solve the matrix equations:

$$\begin{aligned} H &= B_0 + B_1 H + B_2 H^2 \\ K &= B_0 K^2 + B_1 K + B_2 \\ T &= T^2 B_0 + T B_1 + B_2 \\ V &= B_0 + V B_1 + V^2 B_2. \end{aligned} \tag{12}$$

Proof. First observe that if $\boldsymbol{u}$ is positive then $\boldsymbol{u}^T S\boldsymbol{e} < 1$, since $S\boldsymbol{e} \leq \boldsymbol{e}$ and S is substochastic [10, 14], hence $I - \boldsymbol{e}\boldsymbol{u}^T S$ is nonsingular.

The matrix H solves the first equation of (12) by construction. For the matrix K we have:

$$\begin{aligned} &B_0 K^2 + B_1 K + B_2 = \\ &B_0 (I - \boldsymbol{e}\boldsymbol{u}^T S) S^2 (I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} + B_1 (I - \boldsymbol{e}\boldsymbol{u}^T S) S (I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} + B_2 = \\ &\left(A_0 (I - \boldsymbol{e}\boldsymbol{u}^T) S^2 + \left(A_1 + (A_0 \boldsymbol{e} - \boldsymbol{e})\boldsymbol{u}^T S + A_2 \boldsymbol{e}\boldsymbol{u}^T \right) S + \right. \\ &\left. + A_2 (I - \boldsymbol{e}\boldsymbol{u}^T S) \right) (I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} = \\ &(S - \boldsymbol{e}\boldsymbol{u}^T S^2)(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1} = K. \end{aligned}$$

For the matrix $T = R$, since [10] $R = A_2 (I - A_1 - A_2 G)^{-1}$ and

$$\begin{aligned} A_2 \boldsymbol{e} &= R(I - A_1 - R A_0)\boldsymbol{e} = R \left(I - A_1 - A_2 (I - A_1 - A_2 G)^{-1} A_0 \right) \boldsymbol{e} = \\ &\quad R(I - A_1 - A_2)\boldsymbol{e} = R A_0 \boldsymbol{e}, \end{aligned}$$

we obtain

$$\begin{aligned} B_2 + R B_1 + R^2 B_0 &= A_2 + R(A_1 + A_2 \boldsymbol{e}\boldsymbol{u}^T) + R^2 A_0 (I - \boldsymbol{e}\boldsymbol{u}^T) = \\ &\quad A_2 + R A_1 + R^2 A_0 + R A_2 \boldsymbol{e}\boldsymbol{u}^T - R^2 A_0 \boldsymbol{e}\boldsymbol{u}^T = \\ &\quad R + R(A_2 \boldsymbol{e} - R A_0 \boldsymbol{e})\boldsymbol{u}^T = R. \end{aligned}$$

Concerning the matrix V, observe that $I - B_1 - B_0 K$ is nonsingular, since $I - A_1 - A_0 S - B_2 G$ is nonsingular [10, 14], and $I - A_1 - A_0 S - B_2 G =$

$(I - B_1 - B_0K - B_2H)(I - \boldsymbol{e}\boldsymbol{u}^T S) = (I - B_1 - B_0K)(I - KH)(I - \boldsymbol{e}\boldsymbol{u}^T S)$. It can be shown by direct substitution that $V = B_0(I - B_1 - B_0K)^{-1}$ solves the last matrix equation in (12). By rewriting the matrix V in terms of the blocks A_0, A_1, A_2 we obtain that:

$$V = A_0(I - \boldsymbol{e}\boldsymbol{u}^T)(I - A_1 - A_0S - A_2\boldsymbol{e}\boldsymbol{u}^T)^{-1}.$$

On the other hand it holds:

$$\begin{aligned}
&(I - A_1 - A_0S - A_2\boldsymbol{e}\boldsymbol{u}^T)^{-1} = \\
&(I - A_1 - A_0S)^{-1} + \\
&(1 - \boldsymbol{u}^T(I - A_1 - A_0S)^{-1}A_2\boldsymbol{e})^{-1}(I - A_1 - A_0S)^{-1}A_2\boldsymbol{e}\boldsymbol{u}^T(I - A_1 - A_0S)^{-1} = \\
&(I - A_1 - A_0S)^{-1} + (1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1}S\boldsymbol{e}\boldsymbol{u}^T(I - A_1 - A_0S)^{-1} = \\
&\left(I - (1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1}S\boldsymbol{e}\boldsymbol{u}^T\right)(I - A_1 - A_0S)^{-1}.
\end{aligned}$$

From the latter equation and from the relation $F = A_0(I - A_1 - A_0S)^{-1}$ [10], we obtain:

$$\begin{aligned}
V \;=\; & F + (1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1}A_0S\boldsymbol{e}\boldsymbol{u}^T(I - A_1 - A_0S)^{-1} - \\
& - \left(A_0\boldsymbol{e} + (\boldsymbol{u}^T S\boldsymbol{e})(1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1}A_0\boldsymbol{e}\right)\boldsymbol{u}^T(I - A_1 - A_0S)^{-1} = \\
& F - \boldsymbol{w}\boldsymbol{u}^T(I - A_1 - A_0S)^{-1},
\end{aligned}$$

where $\boldsymbol{w} = (1 - \boldsymbol{u}^T S\boldsymbol{e})^{-1}A_0(I - S)\boldsymbol{e}$. □

The spectral properties of the solutions of the shifted matrix equations allow us to prove that the rate of convergence is improved, with respect to standard cyclic reduction. Let us define

$$\begin{aligned}
&\bar{\theta} = \max\{|\lambda| : \ |\lambda| < 1, \ \phi(\lambda) = 0\} \\
&\sigma = \min\{|\lambda| : \ |\lambda| > 1, \ \phi(\lambda) = 0\}.
\end{aligned} \tag{13}$$

The convergence properties that will be proved in the next section rely on the fact that $\bar{\theta} < 1$, $\sigma = 1/\rho(R) > 1$, S and R have the same eigenvalues (see [14]), and thus also K and R, and $\rho(K) = 1/\sigma < 1$.

4 A shifted cyclic reduction algorithm

In this section we apply the cyclic reduction algorithm to solve the shifted matrix equation (7), and we show that the rate of convergence is improved, with respect to the same algorithm applied to the original matrix equation (1).

By following the approach described in section 2, we generate by means of cyclic reduction the sequence of infinite block tridiagonal systems

$$\begin{bmatrix} I-\widehat{B}_1^{(j)} & -B_2^{(j)} & & 0 \\ -B_0^{(j)} & I-B_1^{(j)} & -B_2^{(j)} & \\ & -B_0^{(j)} & I-B_1^{(j)} & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} H \\ H^{2^j+1} \\ H^{2\cdot 2^j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} B_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad j \geq 0, \tag{14}$$

where $B_0^{(0)} = B_0$, $\widehat{B}_1^{(0)} = B_1^{(0)} = B_1$, $B_2^{(0)} = B_2$, and the blocks $\widehat{B}_1^{(j)}$, $B_i^{(j)}$, $i = 0, 1, 2$, $j \geq 1$, are defined by the recurrences:

$$\begin{aligned} B_0^{(j+1)} &= B_0^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_0^{(j)} \\ B_1^{(j+1)} &= B_1^{(j)} + B_0^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_2^{(j)} + B_2^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_0^{(j)} \\ B_2^{(j+1)} &= B_2^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_2^{(j)} \\ \widehat{B}_1^{(j+1)} &= \widehat{B}_1^{(j)} + B_2^{(j)} \left(I - B_1^{(j)}\right)^{-1} B_0^{(j)}. \end{aligned} \tag{15}$$

In the shifted case, the problem of the nonsingularity of the blocks $I - B_1^{(j)}$ must be considered. In fact, in the original case, the matrices $I - A_1^{(j)}$ are nonsingular M-matrices for any j, since $I - A_1^{(j)}$ can be viewed as a Schur complement of the block $(2^{j+1}-1) \times (2^{j+1}-1)$ matrix obtained by truncating the infinite matrix (2) at the block size $2^{j+1} - 1$ (see [1, 6]), and this finite matrix is a nonsingular M-matrix. Analogously, the matrix $I - B_1^{(j)}$ can be viewed as a Schur complement of the matrix $Q_{2^{j+1}-2}$, where Q_n is the $(n+1) \times (n+1)$ block matrix

$$Q_n = \begin{bmatrix} I-B_1 & -B_2 & & 0 \\ -B_0 & \ddots & \ddots & \\ & \ddots & \ddots & -B_2 \\ 0 & & -B_0 & I-B_1 \end{bmatrix}. \tag{16}$$

Thus, the $(j+1)$-st step of cyclic reduction can be performed, i.e., $I - B_1^{(j)}$ is nonsingular, if and only if $Q_{2^{j+1}-2}$ is nonsingular [1]. Based on this property, we prove the following result:

Theorem 4 *Let $\boldsymbol{u}$ be any positive vector such that $\boldsymbol{u}^T\boldsymbol{e} = 1$ and let $H = G - \boldsymbol{u}\boldsymbol{e}^T$, $K = (I - \boldsymbol{e}\boldsymbol{u}^T S)S(I - \boldsymbol{e}\boldsymbol{u}^T S)^{-1}$, where G and S are the minimal nonnegative solutions of the matrix equations $G = A_0 + A_1 G + A_2 G^2$, $S =$*

$A_0S^2 + A_1S + A_2$. *Let* $B_0^{(j)}$, $B_1^{(j)}$, $B_2^{(j)}$, $\widehat{B}_1^{(j)}$ *be the blocks generated at the* j*-th step of cyclic reduction,* $j \geq 1$. *Then, the following conditions are equivalent:*

1. $I - (HK^{2^{j+1}-1})(KH^{2^{j+1}-1})$ *is nonsingular;*
2. $I - (KH^{2^{j+1}-1})(HK^{2^{j+1}-1})$ *is nonsingular;*
3. $I - B_1^{(j)}$ *is nonsingular.*

Proof. The matrix $I - B_1^{(j)}$ is nonsingular if and only if $Q_{2^{j+1}-2}$ is singular, where Q_n is defined in (16). For simplicity of notations, set $n = 2^{j+1}-2$. Observe that the matrix $W = I - B_1 - B_0K - B_2H$ is nonsingular, since $W = (I - A_1 - A_0S - A_2G)(I - \boldsymbol{e}\boldsymbol{u}^TS)$ and $I - A_1 - A_0S - B_2G$ is nonsingular (see [10, 14]). Let $\boldsymbol{x} = (\boldsymbol{x}_i)_{i=0,\ldots,n}$ such that $Q_n\boldsymbol{x} = 0$. Therefore, since $I - B_1 - B_0K - B_2H$ is nonsingular, then

$$\boldsymbol{x}_i = H^i\boldsymbol{r} + K^{n-i}\boldsymbol{s}, \quad i = 0, \ldots, n$$

where $\boldsymbol{r}$ and $\boldsymbol{s}$ are suitable vectors, such that the boundary conditions

$$\begin{array}{l} (I - B_1)\boldsymbol{x}_0 - B_2\boldsymbol{x}_1 = \boldsymbol{0} \\ -B_0\boldsymbol{x}_{n-1} + (I - B_1)\boldsymbol{x}_n = \boldsymbol{0} \end{array} \tag{17}$$

are satisfied (see [15]). By imposing the above equalities, we obtain that $\boldsymbol{r}$ and $\boldsymbol{s}$ must solve the following homogeneous linear system:

$$\begin{bmatrix} I - B_1 - B_2H & ((I - B_1)K - B_2)\,K^{n-1} \\ ((I - B_1)H - B_0)\,H^{n-1} & I - B_1 - B_0K \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{s} \end{bmatrix} = 0,$$

that can be written as

$$\begin{bmatrix} I - B_1 - B_2H & B_0K^{n+1} \\ B_2H^{n+1} & I - B_1 - B_0K \end{bmatrix} \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{s} \end{bmatrix} = 0, \tag{18}$$

since H and K solve the matrix equations (1), (7). Hence, $\boldsymbol{x} = \boldsymbol{0}$ if and only if $\boldsymbol{r} = \boldsymbol{s} = \boldsymbol{0}$. Now, the block diagonal entries in the matrix of (18) are nonsingular since $I - B_1 - B_0K - B_2H = (I - B_1 - B_2H)(I - HK) = (I - B_1 - B_0K)(I - KH)$, and $I - B_1 - B_0K - B_2H$ is nonsingular. By computing the Schur complement of $I - B_1 - B_2H$ in $I - B_1 - B_0K$ we obtain the matrix

$$\begin{aligned} S_1 \;=\; & I - B_1 - B_0K - B_2H^{n+1}(I - B_1 - B_2H)^{-1}B_0K^{n+1} = \\ & I - B_1 - B_0K - B_2H^{n+2}K^{n+1}. \end{aligned}$$

Thus, S_1 (and hence Q_n) is singular if and only if $(I - B_1 - B_0K)^{-1}S_1 = I - KH^{n+2}K^{n+1}$ is nonsingular. By taking the Schur complement of $I - B_1 - B_0K$ in $I - B_1 - B_0K$ we complete the proof of the theorem. □

The above theorem gives necessary and sufficient conditions for the applicability of cyclic reduction. Since $\rho(H) < 1$ and $\rho(K) < 1$, the matrix $Z_j = I - (HK^{2^{j+1}-1})(KH^{2^{j+1}-1})$ is nonsingular for sufficiently large values of j. From the numerical experiments that we have performed, and from the fact that $I - (GS^{2^{j+1}-1})(SG^{2^{j+1}-1})$ is nonsingular for any value of j, we conjecture that also Z_j is nonsingular for any value of j. However, if it were not, the cyclic reduction algorithm can be still applied, by performing a different permutation of block rows and columns, that allows one to skip the steps that cannot be performed for the singularity of the blocks $I - B_1^{(j)}$ (we refer to the paper [1] for details of this subject).

Here and hereafter we assume that the matrices $I - B_1^{(j)}$ are nonsingular for any j.

The nice convergence properties stated by the next theorem enable us to efficiently compute the matrix H:

Theorem 5 *Let $B_0^{(j)}$, $B_1^{(j)}$, $B_2^{(j)}$ be the blocks generated at the j-th step of cyclic reduction. Then $B_1^{(j)}$, $(I - B_1^{(j)})^{-1}$ are bounded, and for any operator norm it holds $||B_2^{(j)}|| = O\left((1/\sigma)^{2^j}\right)$ and $||B_0^{(j)}|| = O\left(\theta^{2^j}\right)$, for any $\bar{\theta} < \theta < 1$, where $\bar{\theta}$ and σ are defined in (13).*

Proof. From (14), at each step j the following equations hold:

$$H^{2^j} = B_0^{(j)} + B_1^{(j)}H^{2^j} + B_2^{(j)}H^{2\cdot 2^j}, \tag{19}$$

$$K^{2^j} = B_0^{(j)}K^{2\cdot 2^j} + B_1^{(j)}K^{2^j} + B_2^{(j)}, \tag{20}$$

and therefore

$$(I - B_1^{(j)})^{-1}B_0^{(j)} + H^{2^j} + (I - B_1^{(j)})^{-1}B_2^{(j)}H^{2\cdot 2^j} = 0, \tag{21}$$

$$(I - B_1^{(j)})^{-1}B_2^{(j)} + K^{2^j} + (I - B_1^{(j)})^{-1}B_0^{(j)}K^{2\cdot 2^j} = 0. \tag{22}$$

We first prove that the matrices $C_0^{(j)} = (I - B_1^{(j)})^{-1}B_0^{(j)}$ and $C_2^{(j)} = (I - B_1^{(j)})^{-1}B_2^{(j)}$ are bounded in norm.

Since the maximum modulus eigenvalue of R, and thus of K, is real, simple and unique [8] then there exists an operator norm $||\cdot||_K$ such that $||K||_K = \rho(K)$. Moreover, let $\epsilon > 0$ such that $\rho(H) + \epsilon < 1$, and let $||\cdot||_{H,\epsilon}$ be an operator norm such that $||H||_{H,\epsilon} \leq \rho(H) + \epsilon$ (see [9]).

Let $\alpha_j = ||C_0^{(j)}||_{H,\epsilon}$. If α_j is not bounded, then there exists a subsequence α_{j_h} such that α_{j_h} diverges to infinity. From (21) we have

$$\begin{aligned}\alpha_{j_h} &\leq ||C_2^{(j_h)}||_{H,\epsilon}||H^{2\cdot 2^{j_h}}||_{H,\epsilon} + ||H^{2^{j_h}}||_{H,\epsilon} \leq \\ &(\rho(H)+\epsilon)^{2^{j_h}} \left((\rho(H)+\epsilon)^{2^{j_h}}||C_2^{(j_h)}||_{H,\epsilon} + 1\right).\end{aligned}$$

Thus,

$$||C_2^{(j_h)}||_{H,\epsilon} \geq \left(\frac{\alpha_{j_h}}{(\rho(H)+\epsilon)^{2^{j_h}}} - 1\right)\frac{1}{(\rho(H)+\epsilon)^{2^{j_h}}}$$

and, since α_{j_h} diverges to infinity, there exists a constant $c > 0$ such that

$$||C_2^{(j_h)}||_{H,\epsilon} \geq c\frac{\alpha_{j_h}}{(\rho(H)+\epsilon)^{2^{j_h}}}.$$

For the equivalence of the operator norms, there exist constants $c' > 0$ and $c'' > 0$ such that

$$||C_2^{(j_h)}||_K \geq c'\frac{\alpha_{j_h}}{(\rho(H)+\epsilon)^{2^{j_h}}}, \tag{23}$$

$$||C_0^{(j)}||_K \leq c''||C_0^{(j)}||_{H,\epsilon} = c''\alpha_j. \tag{24}$$

On the other hand, from (22) we have

$$||C_2^{(j_h)}||_K \leq ||C_0^{(j_h)}||_K||K^{2\cdot 2^{j_h}}||_K + ||K^{2^{j_h}}||_K.$$

Thus, from the latter inequality and (24) we have

$$||C_2^{(j_h)}||_K \leq c''\alpha_{j_h}\rho(K)^{2\cdot 2^{j_h}} + \rho(K)^{2^{j_h}}.$$

Hence, from (23), we obtain

$$c'\frac{\alpha_{j_h}}{(\rho(H)+\epsilon)^{2^{j_h}}} \leq c''\alpha_{j_h}\rho(K)^{2\cdot 2^{j_h}} + \rho(K)^{2^{j_h}},$$

that contradicts the assumption that α_{j_h} goes to infinity. By using a similar argument we can prove that also $C_2^{(j)}$ is bounded in norm. From (21), we obtain

$$||C_0^{(j)}||_{H,\epsilon} \leq (\rho(H)+\epsilon)^{2^j} + ||C_2^{(j)}||_{H,\epsilon}(\rho(H)+\epsilon)^{2\cdot 2^j}.$$

Hence, since $||C_2^{(j)}||_{H,\epsilon}$ is bounded, then there exists a constant $\gamma > 0$ such that

$$||C_0^{(j)}||_{H,\epsilon} \leq \gamma(\rho(H)+\epsilon)^{2^j}.$$

Thus, for the equivalence of the operator norms, for any operator norm $||\cdot||$ there exists a constant γ such that

$$||C_0^{(j)}|| \leq \gamma(\rho(H)+\epsilon)^{2^j}. \tag{25}$$

Similarly, for any operator norm $||\cdot||$ there exists a constant γ' such that

$$||C_2^{(j)}|| \leq \gamma'\rho(K)^{2^j}. \tag{26}$$

From (25) and (26), since

$$I - B_1^{(j+1)} = (I - B_1^{(j)})(I - C_0^{(j)}C_2^{(j)} + C_2^{(j)}C_0^{(j)}),$$

we have

$$||I - B_1^{(j+1)}|| \leq ||I - B_1^{(j)}||(1+\sigma_j)$$

where $\sigma_j = O\left((\rho(H)+\epsilon)^{2^j}\rho(K)^{2^j}\right)$. Thus the matrices $I - B_1^{(j)}$, and hence the matrices $B_1^{(j)}$, are bounded in norm. Similarly, it holds

$$||(I - B_1^{(j+1)})^{-1}|| \leq \frac{1}{1-\sigma_j}||(I - B_1^{(j)})^{-1}||,$$

thus $||(I - B_1^{(j)})^{-1}||$ is bounded.

Now, from the boundness of $||I - B_1^{(j)}||$, and from the relations

$$\begin{aligned} -B_0^{(j)} + (I - B_1^{(j)})H^{2^j} - B_2^{(j)}H^{2\cdot 2^j} &= 0, \\ -B_2^{(j)} + (I - B_1^{(j)})K^{2^j} - B_0^{(j)}K^{2\cdot 2^j} &= 0, \end{aligned}$$

derived from (19) and (20), by applying the same argument used to derive (25) and (26), we can show that $||B_2^{(j)}||$ and $||B_0^{(j)}||$ are bounded, and thus that $||B_0^{(j)}|| = O\left((\rho(H)+\epsilon)^{2^j}\right)$, $||B_2^{(j)}|| = O\left(\rho(K)^{2^j}\right)$. □

The matrix G can be directly recovered, without computing the matrix H, according to the following result:

Theorem 6 *For any operator norm $||\cdot||$ and for any $\bar{\theta} < \theta < 1$ it holds*

$$A_0 - (I - \widehat{B}_1^{(j)})G = O\left(\left(\frac{\theta}{\sigma}\right)^{2^j}\right). \tag{27}$$

Proof. From the relation

$$(I - \widehat{B}_1^{(j)})H - B_2^{(j)}H^{2^j+1} - B_0 = 0$$

and from Theorem 5 it follows that for any operator norm $||\cdot||$ and for any $\bar{\theta} < \theta < 1$ it holds

$$B_0 - (I - \widehat{B}_1^{(j)})H = O\left(\left(\frac{\theta}{\sigma}\right)^{2^j}\right).$$

On the other hand, it can be easily proved by induction that $(I - \widehat{B}_1^{(j)})\boldsymbol{e} = A_0\boldsymbol{e}$ for any $j \geq 0$. Thus, by replacing H with $G - \boldsymbol{e}\boldsymbol{u}^T$ and B_0 with $A_0 - A_0\boldsymbol{e}\boldsymbol{u}^T$, we arrive at (27). □

From the above result, compared with (6), it follows that the shifted cyclic reduction can be much faster than the original one. Indeed, if the second largest modulus eigenvalue $\bar{\theta}$ of G is far from the unit circle, then the rate of convergence is much improved.

Thus, the deflating technique leads to a better rate of convergence, but destroys the nonnegativity and M-matrix properties of the blocks generated at each step. Indeed, in general neither $I - B_1^{(j)}$ is an M-matrix, nor $B_0^{(j)}$, $B_2^{(j)}$ are nonnegative matrices (from the numerical experiments it seems that $I - \widehat{B}_1^{(j)}$ is an M-matrix for $j = 0, 1, \ldots$). In principle this fact could lead to a loss of accuracy of the results obtained with the shifting technique. In practice, we have not observed any differences, in terms of accuracy, between the results obtained with the two algorithms. Furthermore, as we will prove in the next section, the shifted equation is better conditioned than the original one.

5 Conditioning of the shifted matrix equation

In this section we introduce a measure of the conditioning of the matrix equation, and we show that the shifted equation is better conditioned than the original one.

Consider the matrix equation (1) and a perturbed equation

$$G+\triangle G = (A_0+\triangle A_0)+(A_1+\triangle A_1)(G+\triangle G)+(A_2+\triangle A_2)(G+\triangle G)^2. \quad (28)$$

Using (1) this perturbed equation simplifies up to the first oder in $\triangle G$

$$(I - A_1 - A_2G)\triangle G - A_2(\triangle G)G = \triangle A, \quad (29)$$

where

$$\triangle A = \triangle A_0 + (\triangle A_1)G + (\triangle A_2)G^2.$$

Note that (29) can be written

$$W\mathbf{vec}(\triangle G) = \mathbf{vec}(\triangle A), \tag{30}$$

where

$$W = I \otimes (I - A_1 - A_2G) + G^T \otimes (-A_2)$$

and $\mathbf{vec}(A)$ is the k^2-dimensional vector obtained by column-wise arranging the entries of the matrix A.

Let $\Lambda(M)$ denote the set of all eigenvalues of any square matrix M.

Theorem 7 *The $k^2 \times k^2$ matrix W is nonsingular. Furthermore*

$$\min\{|\lambda| : \lambda \in \Lambda(W)\} = 1 - \rho(A_1 + A_2G + A_2).$$

Proof. Let S be the Schur canonical form of G^T. Then the matrix W is similar to

$$\tilde{W} = I \otimes (I - A_1 - A_2G) + S \otimes (-A_2).$$

Due to the upper triangular structure of S and the fact that $\Lambda(S) = \Lambda(G)$ it is easy to see that

$$\Lambda(W) = \Lambda(\tilde{W}) = \bigcup_{\lambda \in \Lambda(G)} \Lambda(I - A_1 - A_2G - \lambda A_2).$$

Note that whenever $\lambda \in \Lambda(G)$

$$\begin{aligned} \rho(A_1 + A_2G + \lambda A_2) &\leq \rho(|A_1 + A_2G + \lambda A_2|) \\ &\leq \rho(A_1 + A_2G + |\lambda| A_2) \\ &\leq \rho(A_1 + A_2G + A_2) \\ &< 1. \end{aligned}$$

In the last step we used the fact that since $A_0 + A_1 + A_2$ is irreducible and positive recurrent the matrix $A_1 + A_2G + A_2$ has spectral radius less than 1. It follows that 0 does not belong to $\Lambda(W)$, and hence W is nonsingular. Clearly

$$\min\{|\lambda| : \lambda \in \Lambda(W)\} = 1 - \rho(A_1 + A_2G + A_2)$$

□

Since W is nonsingular

$$\mathbf{vec}(\triangle G) = W^{-1}\mathbf{vec}(\triangle A).$$

Hence

$$\begin{aligned}
\|\triangle G\|_F &= \|\mathbf{vec}(\triangle G)\|_2 \\
&\leq \|W^{-1}\|_2 \|\mathbf{vec}(\triangle A)\|_2 \\
&= \frac{1}{\sigma_{min}(W)} \|\triangle A_0 + (\triangle A_1)G + (\triangle A_2)G^2\|_F \\
&\leq \frac{\sqrt{k}}{\sigma_{min}(W)} (\|\triangle A_0\|_F + \|\triangle A_1\|_F + \|\triangle A_2\|_F).
\end{aligned}$$

Here $\sigma_{min}(W)$ is the minimum singular value of the matrix W. So we may view $1/\sigma_{min}(W)$ as a condition number of the equation (1). Even though $1/\min\{|\lambda| : \lambda \in \Lambda(W)\} \leq 1/\sigma_{min}(W)$, we may consider

$$\frac{1}{\min\{|\lambda| : \lambda \in \Lambda(W)\}} = \frac{1}{1 - \rho(A_1 + A_2G + A_2)}$$

as a number which reflects the conditioning of the equation (1).

Consider now the shifted matrix equation (7) and the perturbed equation

$$H + \triangle H = (B_0 + \triangle B_0) + (B_1 + \triangle B_1)(H + \triangle H) + (B_2 + \triangle B_2)(H + \triangle H)^2.$$

As before this perturbed equation simplifies up to the first oder in $\triangle H$

$$(I - B_1 - B_2H)\triangle H - B_2(\triangle H)H = \triangle B, \tag{31}$$

where

$$\triangle B = \triangle B_0 + (\triangle B_1)H + (\triangle B_2)H^2.$$

Note that (31) can be written

$$Q\mathbf{vec}(\triangle H) = \mathbf{vec}(\triangle B),$$

where

$$Q = I \otimes (I - B_1 - B_2H) + H^T \otimes (-B_2).$$

Note that

$$B_1 + B_2H = A_1 + A_2\boldsymbol{e}\boldsymbol{u}^T + A_2(G - \boldsymbol{e}\boldsymbol{u}^T) = A_1 + A_2G.$$

Hence

$$Q = I \otimes (I - A_1 - A_2G) + H^T \otimes (-A_2).$$

Note that if we denote $\Lambda(G)$ by

$$\Lambda(G) = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$$

with

$$1 = \lambda_1 > |\lambda_2| \geq \ldots \geq |\lambda_k|$$

then

$$\Lambda(H) = \{\lambda_2, \lambda_3, \ldots, \lambda_k, 0\}$$

and $|\lambda_2| = \bar{\theta}$.

Theorem 8 *The $k^2 \times k^2$ matrix Q is nonsingular. Furthermore*

$$\min\{|\lambda| : \lambda \in \Lambda(Q)\} \geq 1 - \rho(A_1 + A_2G + \bar{\theta}A_2)$$

and the equality holds if λ_2 is a positive real number.

Proof. As in the proof of Theorem 7 it is easy to see that

$$\Lambda(Q) = \Lambda(\tilde{Q}) = \bigcup_{\lambda \in \Lambda(H)} \Lambda(I - A_1 - A_2G - \lambda A_2).$$

Note that since $\rho(H) = \bar{\theta} = |\lambda_2|$ whenever $\lambda \in \Lambda(H)$

$$\begin{aligned} \rho(A_1 + A_2G + \lambda A_2) &\leq \rho(|A_1 + A_2G + \lambda A_2|) \\ &\leq \rho(A_1 + A_2G + |\lambda| A_2) \\ &\leq \rho(A_1 + A_2G + |\lambda_2| A_2) \\ &< 1. \end{aligned}$$

In the last step we used the fact that

$$\rho(A_1 + A_2G + |\lambda_2|A_2) < \rho(A_1 + A_2G + A_2) < 1.$$

It follows that 0 does not belong to $\Lambda(Q)$, and hence Q is nonsingular. Clearly

$$\min\{|\lambda| : \lambda \in \Lambda(Q)\} \geq 1 - \rho(A_1 + A_2G + |\lambda_2|A_2)$$

and the equality holds if λ_2 is a positive real number. □

Since Q is nonsingular

$$\mathbf{vec}(\triangle H) = Q^{-1}\mathbf{vec}(\triangle B).$$

Hence

$$\begin{aligned}\|\triangle H\|_F &= \|\mathbf{vec}(\triangle H)\|_2 \\ &\leq \|Q^{-1}\|_2\|\mathbf{vec}(\triangle B)\|_2 \\ &= \frac{1}{\sigma_{min}(Q)}\|\triangle B_0 + (\triangle B_1)H + (\triangle B_2)H^2\|_F \\ &\leq \frac{2\sqrt{k}}{\sigma_{min}(Q)}(\|\triangle B_0\|_F + \|\triangle B_1\|_F + \|\triangle B_2\|_F).\end{aligned}$$

Here $\sigma_{min}(Q)$ is the minimum singular value of the matrix Q. So we may view $1/\sigma_{min}(Q)$ as a condition number of the shifted equation (7). Even though $1/\min\{|\lambda| : \lambda \in \Lambda(Q)\} \leq 1/\sigma_{min}(Q)$, we may consider $1/\min\{|\lambda| : \lambda \in \Lambda(Q)\}$ as a number which reflects the conditioning of the shifted equation (7). Since $1/\min\{|\lambda| : \lambda \in \Lambda(Q)\} \leq 1/(1-\rho(A_1 + A_2G + \bar{\theta}A_2))$,

$$\frac{1}{1-\rho(A_1 + A_2G + \bar{\theta}A_2)}$$

is a number which reflects the conditioning of the shifted equation (7). The inequality

$$\frac{1}{1-\rho(A_1 + A_2G + \bar{\theta}A_2)} < \frac{1}{1-\rho(A_1 + A_2G + A_2)},$$

suggests that the shifted equation (7) has a better conditioning than the original equation (1).

6 Numerical results

We have tested the cyclic reduction algorithm and the shifted cyclic reduction algorithm on the examples in [13] using Matlab. In the case of the cyclic reduction algorithm we stopped when

$$\|\widehat{A}_1^{(j)} - \widehat{A}_1^{(j-1)}\|_\infty \leq 10^{-12}$$

and we accepted

$$\tilde{G} = \left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$$

as an approximation of G. Similarly, in the case of a shifted cyclic reduction we stopped when

$$\|\widehat{B}_1^{(j)} - \widehat{B}_1^{(j-1)}\|_\infty \leq 10^{-12}$$

	Cyclic Reduction			Shifted Cyclic Reduction		
p	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	9	0.0	$6.7 \cdot 10^{-16}$	2	0.0	$2.2 \cdot 10^{-16}$
10^{-2}	13	0.0	$7.9 \cdot 10^{-15}$	2	0.0	$1.1 \cdot 10^{-16}$
10^{-3}	16	0.0	$6.0 \cdot 10^{-15}$	2	0.0	0.0
10^{-4}	20	$1.1 \cdot 10^{-16}$	$3.4 \cdot 10^{-13}$	2	0.0	0.0
10^{-5}	23	$2.2 \cdot 10^{-16}$	$1.3 \cdot 10^{-11}$	2	$2.2 \cdot 10^{-16}$	0.0
10^{-6}	26	$2.2 \cdot 10^{-16}$	$4.9 \cdot 10^{-11}$	2	$2.2 \cdot 10^{-16}$	$1.1 \cdot 10^{-16}$
10^{-7}	30	$2.2 \cdot 10^{-16}$	$3.1 \cdot 10^{-10}$	2	$5.6 \cdot 10^{-17}$	$2.2 \cdot 10^{-16}$
10^{-8}	33	0.0	$7.9 \cdot 10^{-9}$	2	0.0	0.0
10^{-9}	36	$1.1 \cdot 10^{-16}$	$3.0 \cdot 10^{-8}$	2	$5.6 \cdot 10^{-17}$	$1.1 \cdot 10^{-16}$
10^{-10}	39	0.0	$2.8 \cdot 10^{-7}$	2	0.0	0.0

Table 1: Example 1

and we accepted

$$\tilde{G} = \left(I - \widehat{B}_1^{(j)}\right)^{-1} A_0$$

as an approximation of G. For each example we have reported tables with the number of iterations, the residual error, Res., which is defined by $\|\tilde{G} - A_0 - A_1\tilde{G} - A_2\tilde{G}^2\|_\infty$, and the closeness to stochasticity, Stoc., which is defined by $\|\tilde{G}\boldsymbol{e} - \boldsymbol{e}\|_\infty$, for standard and shifted cyclic reduction. For particular examples we have also reported tables with the values of $\bar{\theta}$, of σ, $1/\sigma$ and $\bar{\theta}/\sigma$, to show the reduction of the rate of convergence, and tables with $1-\rho(A_1+A_2G+A_2)$, $1-\rho(A_1+A_2G+\bar{\theta}A_2)$, $\sigma_{min}(W)$, $\sigma_{min}(Q)$, that show the conditioning of the original and shifted matrix equations.

Example 1 Here the blocks, having dimension 2, depend on a parameter $p > 0$, and are defined as

$$A_0 = \begin{bmatrix} 1-p & 0 \\ 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & p \\ 2p & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1-2p \end{bmatrix}$$

We have tested with ten different p values and obtained the following results summarized in Tables 1, 2, and 3.

Example 2 In this example, A_0, A_1, and A_2 are defined as follows. First we define 24×24 matrices A_0', A_1', and A_2' as follows:

$$(A_0') = \begin{cases} 192(1 - i/24), & i = j \\ 0, & i \neq j \end{cases} \qquad (A_2') = \begin{cases} 192\rho_d, & i = j \\ 0, & i \neq j \end{cases}$$

p	$\tilde{\theta}$	σ	$1/\sigma$	$\tilde{\theta}/\sigma$
10^{-1}	0	1.125	0.8889	0
10^{-2}	0	1.0102	0.9899	0
10^{-3}	0	1.001	0.9990	0
10^{-4}	0	1.0001	0.9999	0
10^{-5}	0	1.00001	0.99999	0

Table 2: Example 1

p	$1-\rho(A_1+A_2G+A_2)$	$1-\rho(A_1+A_2G+\tilde{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
10^{-1}	0.09	0.68	0.05	0.35
10^{-6}	$1.0 \cdot 10^{-6}$	0.999	$5.0 \cdot 10^{-7}$	0.30
10^{-10}	$1.0 \cdot 10^{-10}$	1.0	$5.0 \cdot 10^{-11}$	0.44

Table 3: Example 1

and

$$(A_1') = \begin{cases} ar(M-i)/M, & i-j=-1 \\ ir, & i-j=1 \\ 0, & \text{elsewhere} \end{cases}$$

Here i and j are integers between 0 and 23, and a, r, M, and ρ_d are parameters. Now A_0, A_1, and A_2 are defined such that $A_0 = sA_0'$, $A_1 = sA_1'$, and $A_2 = sA_2'$ where s is a scalar which makes $A_0 + A_1 + A_2$ stochastic. Tables 4, 5, and 6 report the results obtained by choosing different values of M while we fix $r = 1/300$, $a = 18.244$, and $\rho_d = 0.280$. Tables 7, 8, and 9 report the results obtained by choosing different values of ρ_d while we fix $r = 1/100$, $a = 18.244$, and $M = 512$.

Example 3 In this example we construct a QBD problem defined by the $k \times k$ matrices $A_0 = R + \delta I$, $A_1 = A_2 = R$, where R is a matrix having null diagonal entries and constant off-diagonal entries, and $0 < \delta < 1$. As was observed in [13], the rate $\rho = p^T(A_1 + 2A_2)e$, where $p^T(A_0 + A_1 + A_2) = p^T$, $p^T e = 1$, is exactly $1-\delta$. We have tested with eight different δ values. Tables 10, 11, and 12 report the results obtained with $k = 16$ and Tables 13 and 14 report the results obtained with sizes $k = 32$ and $k = 64$, respectively.

	Cyclic Reduction			Shifted Cyclic Reduction		
M	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
64	19	$1.6 \cdot 10^{-16}$	$3.2 \cdot 10^{-12}$	18	$5.1 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
128	20	$2.2 \cdot 10^{-16}$	$8.4 \cdot 10^{-13}$	19	$6.9 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
256	21	$3.0 \cdot 10^{-16}$	$5.9 \cdot 10^{-12}$	19	$4.6 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
512	22	$2.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-12}$	19	$4.7 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
1024	23	$2.4 \cdot 10^{-16}$	$2.1 \cdot 10^{-11}$	19	$5.3 \cdot 10^{-16}$	$5.6 \cdot 10^{-16}$
2048	24	$3.0 \cdot 10^{-16}$	$6.1 \cdot 10^{-11}$	19	$5.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
4096	25	$2.6 \cdot 10^{-16}$	$5.4 \cdot 10^{-11}$	19	$5.0 \cdot 10^{-17}$	$6.7 \cdot 10^{-16}$
8192	26	$3.1 \cdot 10^{-16}$	$3.9 \cdot 10^{-10}$	19	$4.8 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
16384	27	$2.2 \cdot 10^{-16}$	$2.0 \cdot 10^{-11}$	19	$5.9 \cdot 10^{-17}$	$6.7 \cdot 10^{-16}$
32768	29	$1.3 \cdot 10^{-16}$	$1.1 \cdot 10^{-9}$	19	$3.5 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
65536	34	$2.2 \cdot 10^{-16}$	$5.5 \cdot 10^{-8}$	19	$5.0 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$

Table 4: Example 2: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$

M	$\bar{\theta}$	σ	$1/\sigma$	$\bar{\theta}/\sigma$
64	0.9999	1.0002	0.9998	0.9997
128	0.9999	1.0001	0.9999	0.9998
256	0.9999	1.000039	0.999961	0.9998
512	0.9999	1.000019	0.999981	0.9998
1024	0.9999	1.000009	0.999991	0.9998
2048	0.9999	1.000004	0.999996	0.9998
4096	0.9999	1.000002	0.999998	0.9999
8192	0.9999	1.000001	0.999999	0.9999
16384	0.9999	1.0000004	0.9999996	0.9999
32768	0.9999	1.0000001	0.9999999	0.9999
65536	0.9999	1.000000002	0.999999998	0.9999

Table 5: Example 2: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$

M	$1 - \rho(A_1 + A_2G + A_2)$	$1 - \rho(A_1 + A_2G + \bar{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
64	$1.4 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$	$2.7 \cdot 10^{-5}$	$3.8 \cdot 10^{-5}$
8192	$7.1 \cdot 10^{-7}$	$1.1 \cdot 10^{-4}$	$1.5 \cdot 10^{-7}$	$1.5 \cdot 10^{-5}$
65536	$1.6 \cdot 10^{-9}$	$1.0 \cdot 10^{-4}$	$3.5 \cdot 10^{-10}$	$1.5 \cdot 10^{-5}$

Table 6: Example 2: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$

	Cyclic Reduction			Shifted Cyclic Reduction		
ρ_d	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
0.01	6	$2.2 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$	6	$3.9 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$
0.025	7	$3.3 \cdot 10^{-16}$	$6.6 \cdot 10^{-16}$	7	$5.7 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.05	10	$1.5 \cdot 10^{-16}$	$8.9 \cdot 10^{-16}$	10	$3.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.075	12	$2.2 \cdot 10^{-16}$	$9.0 \cdot 10^{-15}$	12	$6.1 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.1	14	$2.2 \cdot 10^{-16}$	$2.2 \cdot 10^{-15}$	14	$4.2 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.12	14	$2.2 \cdot 10^{-16}$	$2.8 \cdot 10^{-14}$	14	$3.9 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.14	15	$2.2 \cdot 10^{-16}$	$2.1 \cdot 10^{-13}$	15	$5.1 \cdot 10^{-17}$	$4.4 \cdot 10^{-16}$
0.16	16	$2.2 \cdot 10^{-16}$	$1.5 \cdot 10^{-14}$	16	$5.3 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.18	16	$2.2 \cdot 10^{-16}$	$5.9 \cdot 10^{-14}$	16	$4.7 \cdot 10^{-17}$	$3.3 \cdot 10^{-16}$
0.2	17	$2.2 \cdot 10^{-16}$	$1.0 \cdot 10^{-14}$	16	$4.5 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
0.22	18	$2.2 \cdot 10^{-16}$	$4.3 \cdot 10^{-13}$	17	$6.0 \cdot 10^{-16}$	$3.3 \cdot 10^{-16}$
0.24	18	$2.3 \cdot 10^{-16}$	$7.8 \cdot 10^{-13}$	17	$3.7 \cdot 10^{-16}$	$6.7 \cdot 10^{-16}$
0.26	19	$1.9 \cdot 10^{-16}$	$8.6 \cdot 10^{-13}$	17	$5.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$
0.28	20	$2.3 \cdot 10^{-16}$	$3.5 \cdot 10^{-12}$	17	$5.1 \cdot 10^{-16}$	$3.3 \cdot 10^{-16}$
0.29	22	$2.6 \cdot 10^{-16}$	$1.8 \cdot 10^{-11}$	17	$3.7 \cdot 10^{-16}$	$2.2 \cdot 10^{-16}$
.29568	31	$2.6 \cdot 10^{-16}$	$1.9 \cdot 10^{-8}$	17	$6.0 \cdot 10^{-16}$	$4.4 \cdot 10^{-16}$

Table 7: Example 2: $r = 1/100,\ a = 18.244,\ M = 512$

ρ_d	$\bar{\theta}$	σ	$1/\sigma$	$\bar{\theta}/\sigma$
0.01	0.9998	4.3182	0.2316	0.2315
0.025	0.9998	1.7714	0.5645	0.5644
0.05	0.9998	1.0814	0.9248	0.9246
0.075	0.9998	1.0170	0.9833	0.9831
0.1	0.9998	1.0063	0.9938	0.9936
0.12	0.9998	1.0034	0.9966	0.9964
0.14	0.9997	1.0021	0.9979	0.9977
0.16	0.9997	1.0013	0.9987	0.9984
0.18	0.9997	1.0009	0.9991	0.9988
0.2	0.9997	1.0006	0.9994	0.9991
0.22	0.9997	1.0004	0.9996	0.9993
0.24	0.9996	1.0002	0.9998	0.9994
0.26	0.9996	1.0001	0.9999	0.9995
0.28	0.9996	1.0001	0.9999	0.9995
0.29	0.9996	1.000019	0.99998	0.9995
0.29568	0.9995	1.00000003	0.99999997	0.9995

Table 8: Example 2: $r = 1/100,\ a = 18.244,\ M = 512$

p	$1-\rho(A_1+A_2G+A_2)$	$1-\rho(A_1+A_2G+\bar{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
0.01	0.6265	0.6265	0.6259	0.6175
0.28	$4.0\cdot 10^{-5}$	$3.4\cdot 10^{-4}$	$8.7\cdot 10^{-6}$	$5.3\cdot 10^{-5}$
0.29568	$1.9\cdot 10^{-8}$	$3.2\cdot 10^{-4}$	$3.6\cdot 10^{-9}$	$4.4\cdot 10^{-5}$

Table 9: Example 2: $r = 1/100,\ a = 18.244,\ M = 512$

	Cyclic Reduction			Shifted Cyclic Reduction		
δ	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	8	$5.7\cdot 10^{-16}$	$1.3\cdot 10^{-15}$	5	$2.8\cdot 10^{-16}$	$3.3\cdot 10^{-16}$
10^{-2}	11	$6.5\cdot 10^{-16}$	$7.2\cdot 10^{-15}$	4	$3.7\cdot 10^{-16}$	0.0
10^{-3}	14	$6.7\cdot 10^{-16}$	$9.7\cdot 10^{-14}$	4	$1.9\cdot 10^{-16}$	$2.2\cdot 10^{-16}$
10^{-4}	17	$6.4\cdot 10^{-16}$	$8.8\cdot 10^{-13}$	4	$4.5\cdot 10^{-16}$	$2.2\cdot 10^{-16}$
10^{-5}	21	$1.1\cdot 10^{-15}$	$1.2\cdot 10^{-11}$	4	$3.5\cdot 10^{-16}$	$4.4\cdot 10^{-16}$
10^{-6}	24	$6.5\cdot 10^{-16}$	$5.8\cdot 10^{-11}$	4	$2.5\cdot 10^{-16}$	$8.9\cdot 10^{-16}$
10^{-7}	27	$6.3\cdot 10^{-16}$	$1.4\cdot 10^{-9}$	4	$2.5\cdot 10^{-16}$	$6.7\cdot 10^{-16}$
10^{-8}	29	$7.0\cdot 10^{-16}$	$3.5\cdot 10^{-9}$	4	$2.5\cdot 10^{-16}$	$4.4\cdot 10^{-16}$

Table 10: Example 3: $k = 16$

δ	$\bar{\theta}$	σ	$1/\sigma$	$\bar{\theta}/\sigma$
10^{-1}	0.0783	1.3333	0.75	0.0587
10^{-2}	0.0174	1.0303	0.9706	0.0140
10^{-3}	0.0207	1.0030	0.9970	0.0207
10^{-4}	0.0216	1.0003	0.9997	0.0216
10^{-5}	0.0217	1.00003	0.99997	0.0217

Table 11: Example 3: $k = 16$

δ	$1-\rho(A_1+A_2G+A_2)$	$1-\rho(A_1+A_2G+\bar{\theta}A_2)$	$\sigma_{min}(W)$	$\sigma_{min}(Q)$
10^{-1}	0.1	0.3765	0.1	0.3234
10^{-5}	$1.0\cdot 10^{-5}$	0.3261	$1.0\cdot 10^{-5}$	0.2310
10^{-8}	$1.0\cdot 10^{-8}$	0.3261	$1.0\cdot 10^{-8}$	0.2721

Table 12: Example 3: $k = 16$

	Cyclic Reduction			Shifted Cyclic Reduction		
δ	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	8	$1.3 \cdot 10^{-15}$	$1.1 \cdot 10^{-15}$	5	$1.2 \cdot 10^{-15}$	$8.9 \cdot 10^{-16}$
10^{-2}	11	$9.5 \cdot 10^{-16}$	$1.1 \cdot 10^{-14}$	4	$4.1 \cdot 10^{-16}$	$7.8 \cdot 10^{-16}$
10^{-3}	14	$1.4 \cdot 10^{-15}$	$8.0 \cdot 10^{-14}$	4	$5.4 \cdot 10^{-16}$	$1.1 \cdot 10^{-15}$
10^{-4}	17	$1.6 \cdot 10^{-15}$	$3.7 \cdot 10^{-12}$	4	$6.8 \cdot 10^{-16}$	$1.8 \cdot 10^{-15}$
10^{-5}	21	$1.1 \cdot 10^{-15}$	$1.0 \cdot 10^{-11}$	4	$5.1 \cdot 10^{-16}$	$6.6 \cdot 10^{-16}$
10^{-6}	24	$1.2 \cdot 10^{-15}$	$2.1 \cdot 10^{-10}$	4	$5.5 \cdot 10^{-16}$	$8.9 \cdot 10^{-16}$
10^{-7}	27	$1.1 \cdot 10^{-15}$	$7.9 \cdot 10^{-10}$	4	$6.5 \cdot 10^{-16}$	$1.0 \cdot 10^{-15}$
10^{-8}	29	$8.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-8}$	4	$7.1 \cdot 10^{-16}$	$7.8 \cdot 10^{-16}$

Table 13: Example 3: $k = 32$

	Cyclic Reduction			Shifted Cyclic Reduction		
δ	Iter.	Res.	Stoc.	Iter.	Res.	Stoc.
10^{-1}	8	$2.8 \cdot 10^{-15}$	$4.9 \cdot 10^{-15}$	5	$2.0 \cdot 10^{-15}$	$1.8 \cdot 10^{-15}$
10^{-2}	11	$3.1 \cdot 10^{-15}$	$1.1 \cdot 10^{-14}$	4	$1.4 \cdot 10^{-15}$	$2.7 \cdot 10^{-15}$
10^{-3}	14	$3.1 \cdot 10^{-15}$	$1.8 \cdot 10^{-13}$	4	$1.3 \cdot 10^{-15}$	$1.1 \cdot 10^{-15}$
10^{-4}	17	$2.4 \cdot 10^{-15}$	$4.6 \cdot 10^{-12}$	4	$1.2 \cdot 10^{-15}$	$2.9 \cdot 10^{-15}$
10^{-5}	21	$2.3 \cdot 10^{-15}$	$2.9 \cdot 10^{-11}$	4	$1.6 \cdot 10^{-15}$	$4.4 \cdot 10^{-16}$
10^{-6}	24	$3.3 \cdot 10^{-15}$	$2.8 \cdot 10^{-10}$	4	$1.4 \cdot 10^{-15}$	$2.4 \cdot 10^{-15}$
10^{-7}	27	$2.4 \cdot 10^{-15}$	$1.9 \cdot 10^{-10}$	4	$1.3 \cdot 10^{-15}$	$3.8 \cdot 10^{-15}$
10^{-8}	29	$2.6 \cdot 10^{-15}$	$1.7 \cdot 10^{-8}$	4	$1.2 \cdot 10^{-15}$	$3.3 \cdot 10^{-15}$

Table 14: Example 3: $k = 64$

References

[1] D. A. Bini, L. Gemignani, and B. Meini. Factorization of analytic functions by means of Koenig's theorem and Toeplitz computations. *Numer. Math.*, 2000. To appear.

[2] D. A. Bini and B. Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 21–38. Kluwer Academic Publisher, 1995.

[3] D. A. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:906–926, 1996.

[4] D. A. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 15:57–74, 1997.

[5] D. A. Bini and B. Meini. Inverting block Toeplitz matrices in block Hessenberg form by means of displacement operators: application to queueing problems. *Linear Algebra Appl.*, 272:1–16, 1998.

[6] D. A. Bini and B. Meini. Effective methods for solving banded Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 20:700–719, 1999.

[7] H. R. Gail, S. L. Hantler, and B. A. Taylor. Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains. *Adv. in Appl. Probab.*, 28:114–165, 1996.

[8] H. R. Gail, S. L. Hantler, and B. A. Taylor. Use of characteristics roots for solving infinite state Markov chains. In W. K. Grassmann, editor, *Computational Probability*, pages 205–255. Kluwer Academic Publishers, 2000.

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

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

[11] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Probability*, 30:650–674, 1993.

[12] G. Latouche and G.W. Stewart. Numerical methods for M/G/1 type queues. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 571–581. Kluwer Academic Publishers, 1995.

[13] B. Meini. Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method. *Adv. Perf. Anal.*, 1:215–225, 1998.

[14] V. Naoumov. Matrix-multiplicative approach to quasi-birth-and-death processes analysis. In A. S. Alfa and S. Chakravarthy, editors, *Matrix-analytic methods in stochastic models (Flint, MI)*, pages 85–106. Dekker, New York, 1997.

[15] V. Naoumov, U. R. Krieger, and D. Wagner. Analysis of a multiserver delay-loss system with a general Markovian arrival process. In A. S. Alfa and S. Chakravarthy, editors, *Matrix-analytic methods in stochastic models (Flint, MI)*, pages 43–66. Dekker, New York, 1997.

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

Solving certain queueing problems by means of regular splittings

Paola Favati* Beatrice Meini†

Abstract

We analyze the problem of the computation of the solution of the nonlinear matrix equation $X = \sum_{i=0}^{+\infty} X^i A_i$, arising in queueing models. We propose a technique based on regular splittings, that one hand leads to a new method for computing the solution, on the other hand it may be used to construct nonlinear matrix equations equivalent to starting one, that can be possibly solved by applying different algorithms.

Key Words. Regular splitting, Markov chain, M/G/1 type matrices.

1 Introduction.

Let P be the infinite column stochastic matrix

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

defined by the $k \times k$ blocks B_{i+1}, A_i, $i \geq 0$. A nonnegative matrix M (denoted with $M \geq O$), possibly infinite, is called column stochastic if $\mathbf{e}^T M = \mathbf{e}^T$, where $\mathbf{e}$ is the vector having all the entries equal to 1. Matrices of the structure (1) are known in literature as stochastic matrices of M/G/1 type [5] and arise in a wide variety of queueing problems modeled by a Markov chain $\mathcal{M}$, where P^T is the transition matrix associated with $\mathcal{M}$. One of the major problems related to Markov chains is the computation of the nonnegative vector $\boldsymbol{\pi}$ such that

$$\boldsymbol{\pi} = P\boldsymbol{\pi}, \quad \mathbf{e}^T\boldsymbol{\pi} = 1. \tag{2}$$

If system (2) has a unique solution, P is called positive recurrent and $\boldsymbol{\pi}$ is called the probability invariant vector associated with P. In the case where P has the structure (1) the computation of

*Istituto di Matematica Computazionale del C.N.R., via S.Maria 46, 56127 Pisa, Italy. E-mail: favati@imc.pi.cnr.it

†Dip. Matematica, via Buonarroti 2, 56127 Pisa, Italy. E-mail: meini@dm.unipi.it

$\boldsymbol{\pi}$ can be reduced (compare [5]) to the computation of the minimal nonnegative solution G of the nonlinear matrix equation

$$X = \sum_{i=0}^{+\infty} X^i A_i, \tag{3}$$

where X is a $k \times k$ matrix. If the matrix P is irreducible and positive recurrent [2], equation (3) has a unique nonnegative solution, which is column stochastic [5].

Once the matrix G is known, an arbitrary number of components of the vector $\boldsymbol{\pi}$ can be recovered by means of a recursive numerically stable formula, called Ramaswami's formula [6], which involves the block entries A_i, B_i of the matrix P in (1), and G.

In this paper we derive a new method for solving the matrix equation (3), that consists in rewriting equation (3) in terms of a linear system and in applying an iterative method, based on regular splittings, for its solution. More precisely, the equation (3) can be rewritten as the following block Toeplitz block Hessenberg infinite system:

$$[G, G^2, G^3, \ldots] \begin{bmatrix} I - A_1 & -A_0 & & \bigcirc & \\ -A_2 & I - A_1 & -A_0 & & \\ -A_3 & -A_2 & I - A_1 & -A_0 & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} = [A_0, O, O, \ldots]\,. \tag{4}$$

In order to solve the above system we generate, by means of regular splittings, a sequence of equivalent systems having block Toeplitz block Hessenberg matrices. We prove that the sequence of transformed systems converges to the system:

$$[G, G^2, G^3, \ldots] \begin{bmatrix} I & -A_0^* & & \bigcirc \\ & I & -A_0^* & \\ \bigcirc & & \ddots & \ddots \end{bmatrix} = [A_0^*, O, O, \ldots], \tag{5}$$

from which we obtain $G = A_0^*$. In this way we generate a sequence of nonlinear matrix equations $X = \sum_{i=0}^{\infty} X^i A_i^{(n)}$, $n \geq 0$, whose solution is still G, that converges to the linear matrix equation $X = A_0^*$, that is immediately solved.

This approach one hand leads to a new method for computing the solution G of (3), on the other hand it may be used to construct nonlinear matrix equations equivalent to (3), that can be possibly solved by applying different algorithms. Indeed, we prove that functional iterations applied to the matrix equation obtained after few steps of the method, converge faster than functional iterations applied to (3), when the initial approximation is the null matrix.

The paper is organized as follows. In Section 2 we analyze the convergence properties of the proposed method, considering also the case where the matrix power series $\sum_{i=0}^{\infty} A_i z^i$ is rational. In Section 3 we relate the regular splittings method with functional iterations, and show some numerical results.

2 The regular splitting method and its convergence properties.

In this section we explain the idea which the method is based on and analyze its convergence properties.

Denoted by H, C and X the coefficient matrix, the right hand side and the unknown of system (4) respectively, we consider the following splitting: $H = M - N$ where

$$M = \begin{bmatrix} I - A_1 & & \bigcirc & \\ -A_2 & I - A_1 & & \\ -A_3 & -A_2 & I - A_1 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \quad N = \begin{bmatrix} O & A_0 & & \bigcirc \\ & O & A_0 & \\ \bigcirc & & \ddots & \ddots \end{bmatrix}$$

obtaining the equivalent system

$$X(I - NM^{-1}) = CM^{-1}. \tag{6}$$

M is a nonsingular matrix such that M^{-1} is nonnegative and N is a nonnegative matrix. Splittings with these properties are called *regular splittings* in [8] and are encountered in the numerical solution of finite Markov chains by means iterative methods (see [7]).

Since M and N are block triangular block Toeplitz matrices, the coefficient matrix $H^{(1)}$ and the right hand side $C^{(1)}$ of the resulting system have the same structure of the initial ones; namely

$$H^{(1)} = I - NM^{-1} = \begin{bmatrix} I - A_1^{(1)} & -A_0^{(1)} & & \bigcirc & \\ -A_2^{(1)} & I - A_1^{(1)} & -A_0^{(1)} & & \\ -A_3^{(1)} & -A_2^{(1)} & I - A_1^{(1)} & -A_0^{(1)} & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

and

$$C^{(1)} = CM^{-1} = \left[A_0^{(1)}, O, O, \ldots\right],$$

where $A_i^{(1)} = A_0 B_{i+1}$, $i = 0, 1, \ldots$, and $B_1 = (I - A_1)^{-1}$, $B_i = (I - A_1)^{-1} \sum_{h=1}^{i-1} A_{i+1-h} B_h$, $i \geq 2$, are the block entries of the first block column of M^{-1}. Moreover, it can be easily verified that $A_i^{(1)} \geq 0, i = 0, 1, \ldots$ and that $\sum_{i=0}^{+\infty} A_i^{(1)}$ is a column stochastic matrix.

So we may iterate this transformation process obtaining a sequence of equivalent systems

$$\left[G, G^2, G^3, \ldots\right] H^{(j)} = \left[A_0^{(j)}, O, O, \ldots\right], \ j \geq 1, \tag{7}$$

with

$$H^{(j)} = \begin{bmatrix} I - A_1^{(j)} & -A_0^{(j)} & & \bigcirc & \\ -A_2^{(j)} & I - A_1^{(j)} & -A_0^{(j)} & & \\ -A_3^{(j)} & -A_2^{(j)} & I - A_1^{(j)} & -A_0^{(j)} & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{8}$$

where the blocks $A_i^{(j)}$ are defined by the recursions:

$$A_i^{(j+1)} = A_0^{(j)} B_{i+1}^{(j)},\ i = 0, 1, \ldots,\ j \geq 1, \tag{9}$$

and

$$B_1^{(j)} = (I - A_1^{(j)})^{-1}, B_i^{(j)} = (I - A_1^{(j)})^{-1} \sum_{h=1}^{i-1} A_{i+1-h}^{(j)} B_h^{(j)},\ i \geq 2,\ j \geq 1, \tag{10}$$

From (7) and (8) it follows that $G = \sum_{i=0}^{\infty} G^i A_i^{(j)}$; in this way at each step of our method we construct a new nonlinear matrix equation having the same nonnegative solution G.

The formulae (9–10) relating the blocks $\{A_i^{(j)}\}_i$ obtained at two subsequent steps can be expressed in functional form in terms of formal matrix power series. Indeed, if we associate with the sequence $\{A_i^{(j)}\}_i$, at step j the formal matrix power series $\varphi^{(j)}(z) = \sum_{i=0}^{\infty} A_i^{(j)} z^i$ we obtain that, for $j \geq 0$,:

$$\varphi^{(j+1)}(z) = \varphi^{(j)}(0)(I - \sum_{i=1}^{\infty} A_i^{(j)} z^{i-1})^{-1} = \varphi^{(j)}(0) \left(I - \frac{\varphi^{(j)}(z) - \varphi^{(j)}(0)}{z} \right)^{-1}. \tag{11}$$

We prove that, if the matrix A_0 is nonsingular, the generated sequence of systems converges to a system that is easy to solve. More precisely, we prove that the sequences $\{H^{(j)}\}_j$ converges to the matrix

$$\begin{bmatrix} I & -A_0^* & & \bigcirc \\ & I & -A_0^* & \\ \bigcirc & & \ddots & \ddots \end{bmatrix}$$

where $A_0^* = \lim_{j\to\infty} A_0^{(j)}$. From (7) we conclude that $G = \lim_{j\to\infty} A_0^{(j)}$. This result is expressed in functional form by the following theorem.

Theorem 1 *Let $\{A_i\}_i$ be a sequence of $k \times k$ nonnegative matrices, such that $\sum_{i=0}^{+\infty} A_i$ is column stochastic and $\det A_0 \neq 0$. Let $\varphi^{(j)}(z) = \sum_{i=0}^{\infty} A_i^{(j)} z^i, j = 1, 2, \ldots$ be defined according to (11),where $\varphi^{(0)}(z) = \sum_{i=0}^{\infty} A_i z^i$, then $\lim_{j\to\infty} \varphi^{(j)}(z) = \lim_{j\to\infty} A_0^{(j)} = A_0^*$ and $A_0^* = G$.*

Proof. First we prove that the $\lim_{j\to\infty} A_0^{(j)}$ exists. Since the following inequalities hold

$$A_0^{(j+1)} = A_0^{(j)} (I - A_1^{(j)})^{-1} \geq A_0^{(j)},\ \ \mathbf{e}^T A_0^{(j)} \leq \mathbf{e}^T, \forall j$$

the sequence $\{A_0^j\}$ is nondecreasing, bounded and hence convergent. Denote with A_0^* the limit matrix.

Then we prove by induction on i that $\lim_{j\to\infty} A_i^{(j)} = 0, i \geq 1$:

- Step 1: $\lim_{j\to\infty} A_1^{(j)} = 0$. Indeed

$$A_0^{(j)} = A_0^{(0)} (I - A_1^{(0)})^{-1} (I - A_1^{(1)})^{-1} \ldots (I - A_1^{(j-1)})^{-1}$$

implies

$$A_0^* = A_0 \prod_{j=0}^{\infty} (I - A_1^{(j)})^{-1}.$$

If $\det A_0 \neq 0$ then $\prod_{j=0}^{\infty}(I - A_1^{(j)})^{-1} = A_0^{-1}A_0^*$ is a convergent product. On the other hand, it is easy to see that

$$\forall n \geq 0, \prod_{j=0}^{n}(I - A_1^{(j)})^{-1} \geq I + \sum_{j=0}^{n} A_1^{(j)},$$

hence also the series $\sum_{j=0}^{\infty} A_1^{(j)}$ is convergent, implying $\lim_{j\to\infty} A_1^{(j)} = 0$.

- Inductive step: assuming that $\lim_{j\to\infty} A_k^{(j)} = 0$, $k = 1, \ldots, i$ we prove that $\lim_{j\to\infty} A_{i+1}^{(j)} = 0$. From relations (9) and (10) we have

$$A_i^{(j+1)} = A_0^{(j)} B_{i+1}^{(j)} = A_0^{(j)}(I - A_1^{(j)})^{-1} \sum_{h=1}^{i} A_{i+2-h}^{(j)} B_h^{(j)}.$$

We isolate in the sum the term containing the matrix $A_{i+1}^{(j)}$, obtaining

$$A_i^{(j+1)} = A_0^{(j+1)} A_{i+1}^{(j)} B_1^{(j)} + A_0^{(j+1)} \sum_{h=2}^{i} A_{i+2-h}^{(j)} B_h^{(j)}. \tag{12}$$

For j going to ∞, the left hand side of (12) goes to 0, by inductive hypotheses; since $A_0^{(j+1)}$ is bounded and $\sum_{h=2}^{i} A_{i+2-h}^{(j)} B_h^{(j)}$ contains matrices $A_k^{(j)}$ with $k \leq i$ the second term in the right hand side of (12) goes to 0, too. Hence

$$\lim_{j\to\infty} A_0^{(j+1)} A_{i+1}^{(j)} B_1^{(j)} = 0. \tag{13}$$

If, by contradiction, $A_{i+1}^{(j)}$ does not converge to the null matrix, a subsequence $\{A_{i+1}^{(j_h)}\}_h$ and a nonnegative matrix R, $R \neq 0$ exist such that $A_{i+1}^{(j_h)} \geq R$, for any h. Then $A_0^{(j_h+1)} A_{i+1}^{(j_h)} B_1^{(j_h)} \geq A_0 A_{i+1}^{(j_h)} \geq A_0 R$, $\forall h$; that is the subsequence $\{A_0^{(j_h+1)} A_{i+1}^{(j_h)} B_1^{(j_h)}\}_h$ is bounded away from zero, being $\det A_0 \neq 0$, and this contradicts (13). □

In the rational case, that is when $\varphi^{(0)}(z)$ is written as a fraction of two polynomial matrices

$$\varphi^{(0)}(z) = P(z)Q(z)^{-1},$$

the sequence $\{\varphi^{(j)}(z)\}_j$ is expressed in the same way

$$\varphi^{(j)}(z) = P^{(j)}(z)Q^{(j)}(z)^{-1}.$$

It is easy to prove that the polynomial matrices associated with $\varphi^{(j+1)}(z)$ and with $\varphi^{(j)}(z)$ are related by the following formulae:

$$\begin{aligned} P^{(j+1)}(z) &= P^{(j)}(0)Q^{(j)}(0)^{-1}Q^{(j)}(z), \\ Q^{(j+1)}(z) &= Q^{(j)}(z) - \tfrac{P^{(j)}(z)-P^{(j+1)}(z)}{z}, \ j \geq 0. \end{aligned} \tag{14}$$

Moreover if $P(z)$ and $Q(z)$ are polynomials of degree p_0 and q_0 respectively, the degrees of the two sequences $\{P^{(j)}(z)\}_j$ and $\{Q^{(j)}(z)\}_j$ do not increase:

$$p_j = \deg(P^{(j)}(z)) \leq \max(p_0, q_0),$$

$$q_j = \deg(Q^{(j)}(z)) \leq \max(p_0, q_0).$$

Hence in the rational case, it is more convenient to use formulae (14), rather than (11), since the degree of $\varphi^{(j)}(z)$ could increase.

3 Regular splittings and functional iteration methods.

In this section we point out some relations between regular splitting method and functional iteration methods. These considerations, together with the results of numerical experimentations, suggest us to apply a few steps of the method, and then to solve the nonlinear matrix equation obtained in this way by functional iteration methods.

The most commonly used functional iteration method are based on the recursion:

$$X_{n+1} = F(X_n), \quad n \geq 0, \tag{15}$$

where X_0 is a nonnegative matrix and $F(\cdot)$ is given by:

$$F(X) = \sum_{i=0}^{+\infty} X^i A_i \tag{16}$$

or

$$F(X) = \left(A_0 + \sum_{i=2}^{+\infty} X^i A_i \right) (I - A_1)^{-1} \tag{17}$$

or

$$F(X) = A_0 \left(I - \sum_{i=1}^{+\infty} X^{i-1} A_i \right)^{-1}. \tag{18}$$

In [3] and [4] it is proved that in the case where $X_0 = 0$ the method based on (18) is faster than the method based on (17) and that the method based on (17) is faster than the method based on (16). This property results experimentally true also when X_0 is a stochastic matrix.

We observe that method based on (17) coincides with method based on (16), applied to solve the matrix equation $X = \sum_{i=0, i\neq 1}^{+\infty} X^i \bar{A}_i$ where $\bar{A}_i = A_i(I - A_1)^{-1}$, $i = 0, 2, 3, \ldots$, are obtained by writing (4) as

$$\left[G, G^2, G^3, \ldots\right] (I - \bar{N}\bar{M}^{-1}) = [A_0, O, O, \ldots] \bar{M}^{-1},$$

where

$$\bar{M} = \begin{bmatrix} I - A_1 & & \bigcirc \\ & I - A_1 & \\ \bigcirc & & \ddots \end{bmatrix}, \quad \bar{N} = \begin{bmatrix} O & A_0 & & \bigcirc \\ A_2 & O & A_0 & \\ A_3 & A_2 & O & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

This splitting cannot be iterated since the blocks on the main diagonal of $\bar{H}^{(1)} = I - \bar{N}\bar{M}^{-1}$ are identity matrices.

Consider now the matrix equation

$$X = \sum_{i=0}^{\infty} X^i A_i^{(1)}, \tag{19}$$

where the blocks $A_i^{(1)}$ are obtained after one step of the regular splitting method described in the previous section. From (11), it follows that the method based on (16), applied to solve (19) is the functional iteration method based on (18), that is the fastest one. Thus, if we apply functional iteration (16) to the matrix equation $X = \sum_{i=0}^{\infty} X^i A_i^{(2)}$, obtained by applying one more step of regular splitting, we have a further improvement of the rate of convergence. This observation suggests us to apply a few steps of regular splittings, and then to apply functional iteration method (16).

We have tested this idea to solve a problem arising in telecommunication modeling, where the size of the blocks A_i is 16, and the number of nonzero blocks is 241 (we refer to [1] for more details). We have applied $h = 1, 5, 10$ regular splittings, and then functional iteration (16), with $X_0 = 0$ and $X_0 = I$. In the following figures we report the logarithm of the residual error $||X_n - \sum_{i=0}^{\infty} X_n^i A_i^{(h)}||_1$ of X_n, versus the number of iterations n, for the different values of h.

From the figures we observe that the asymptotic convergence is improved, as h grows, also in the case $X_0 = I$.

References

[1] G. Anastasi, L. Lenzini, B. Meini, "Performance evaluation of a worst case model of the Metaring MAC Protocol with global fairness", *Performance Evaluation*, 29 (1997), pp. 127–151.

[2] E. Çinlar, Introduction to Stochastic Processes, Prentice-hall, Englewood Cliffs, N.J., 1975.

[3] G. Latouche Algorithms for evaluating the matrix G in Markov chains of PH/G/1 type. Bellcore Technical Report, 1992.

[4] B. Meini, "New convergence results on functional iteration techniques for the numerical solution of M/G/1 type Markov chains", *Numer. Math.*, 78 (1997), pp. 39–58.

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

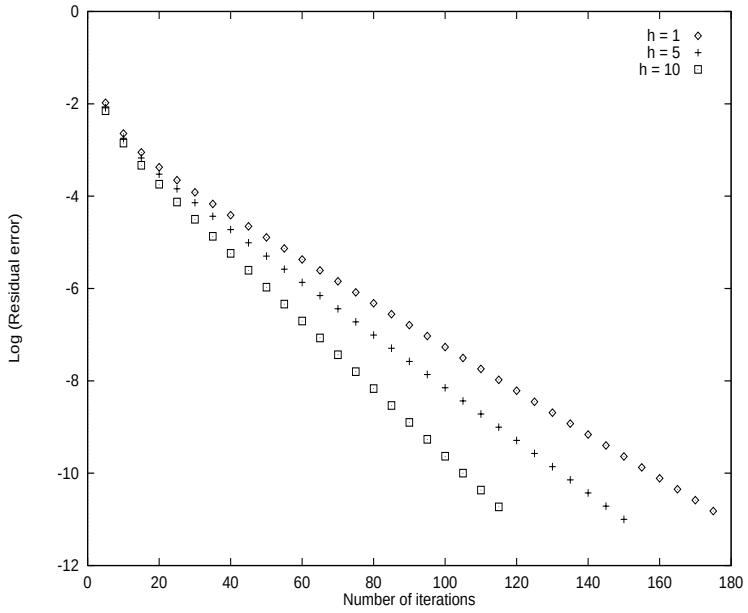


Figure 1: Functional iterations after h regular splittings, $X_0 = 0$

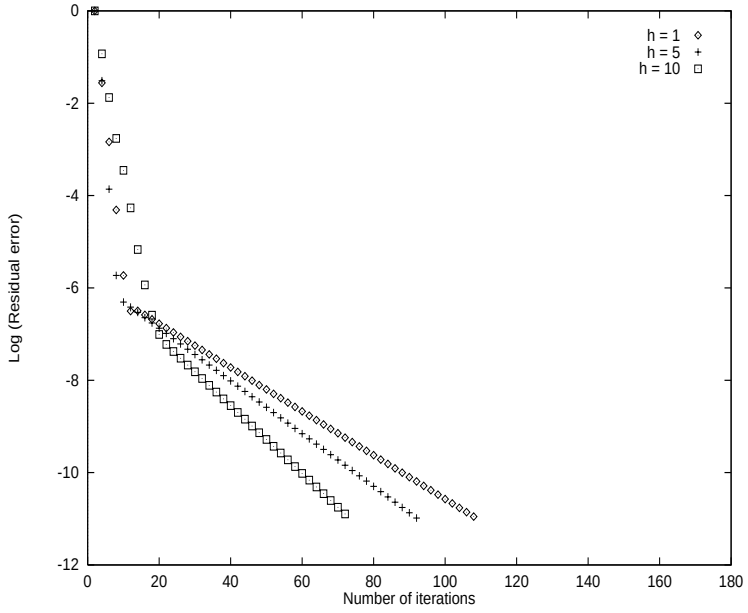


Figure 2: Functional iterations after h regular splittings, $X_0 = I$

[6] V. Ramaswami, "A stable recursion for the steady state vector in Markov chains of M/G/1 type", *Commun. Statist. Stochastic Models,* 4 (1988), pp. 183–188.

[7] D.J. Rose, "Convergent regular splittings for singular M-matrices", *SIAM J. Alg. Discr. Methods,* 5 (1984), pp. 133–144.

[8] R.S. Varga, Matrix Iterative Analysis, New Jersey, Prentice Hall, 1962.

Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method

Beatrice Meini
Dipartimento di Matematica
Università di Pisa - Italy
meini@dm.unipi.it

Abstract

The problem of the computation of the rate matrix G associated with discrete-time QBD Markov chains is analyzed. We present theoretical and numerical comparisons between the cyclic reduction algorithm of [7, 2, 3, 4] and the method based on invariant subspaces of [1].

1 Introduction

The block tridiagonal and block Toeplitz structure of the probability transition matrix P associated with QBD problems, i.e.,

$$P = \begin{bmatrix} B_0 & B_1 & & 0 \\ A_0 & A_1 & A_2 & \\ & A_0 & A_1 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix},$$

allowed the design of fast and reliable methods for the computation of the rate matrix G. Here, A_0, A_1, A_2, B_0, B_1 are $k \times k$ nonnegative matrices such that $A_0+A_1+A_2$ is irreducible, and P is irreducible, stochastic and positive recurrent [8, 9]. These fast methods rely on the property that the matrix G, which solves the nonlinear matrix equation

$$G = A_0 + A_1 G + A_2 G^2, \tag{1}$$

can be computed by solving the infinite block tridiagonal, block Toeplitz system

$$\begin{bmatrix} I-A_1 & -A_2 & & 0 \\ -A_0 & I-A_1 & -A_2 & \\ & -A_0 & I-A_1 & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^2 \\ G^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}. \tag{2}$$

The pioneering method that exploits the Toeplitz structure of the above matrix is the one devised by Latouche and Ramaswami in [7]. In this paper the authors apply a cyclic reduction based method to solve system (2) and derive a quadratically convergent and numerically stable algorithm for the computation of the matrix G. More recently, in [2, 3, 4], Bini and Meini have devised a new quadratically convergent and numerically stable algorithm for the computation of G, based on a functional representation of cyclic reduction, which applies to general M/G/1 type Markov chains [9] and which, in the QBD case, reduces to the method of Latouche and Ramaswami.

A different approach, which apply to M/G/1 type Markov chains having rational generating function (thus in particular to QBD problems), is proposed by Akar and Sohraby in [1]. Here the authors compute the matrix G by finding, by means of matrix sign function iterations, an invariant subspace of a suitable matrix.

The aim of this paper is to compare the cyclic reduction based algorithm of [7, 2, 3, 4] with the approach of [1], in terms of computational cost, numerical stability and convergence rate. Several numerical results show differences in the performance of the two approaches. The cyclic reduction algorithm appears to be more accurate, more robust and faster than the invariant subspace approach.

2 Overview of the two algorithms

Let us first recall the method based on cyclic reduction. For more details we refer to the papers [7, 2, 3, 4].

Let us consider system (2). By recursively applying one step of block cyclic reduction, i.e, an odd-even permutation of block rows and block columns, followed by one step of Gaussian elimination, we generate the sequence of infinite

block tridiagonal systems

$$\begin{bmatrix} I-\widehat{A}_1^{(j)} & -A_2^{(j)} & & 0 \\ -A_0^{(j)} & I-A_1^{(j)} & -A_2^{(j)} & \\ & -A_0^{(j)} & I-A_1^{(j)} & \ddots \\ 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^{2^j+1} \\ G^{2\cdot 2^j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} A_0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad j \geq 0, \tag{3}$$

where $A_0^{(0)} = A_0$, $\widehat{A}_1^{(0)} = A_1^{(0)} = A_1$, $A_2^{(0)} = A_2$. The blocks $\widehat{A}_1^{(j)}$, $A_i^{(j)}$, $i = 0, 1, 2$, $j \geq 1$, are defined by the recurrences:

$$\begin{aligned} A_0^{(j+1)} &= A_0^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)} \\ A_1^{(j+1)} &= A_1^{(j)} + A_0^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_2^{(j)} + A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)} \\ A_2^{(j+1)} &= A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_2^{(j)} \\ \widehat{A}_1^{(j+1)} &= \widehat{A}_1^{(j)} + A_2^{(j)} \left(I - A_1^{(j)}\right)^{-1} A_0^{(j)}, \end{aligned} \tag{4}$$

and are such that $A_0^{(j)} + A_1^{(j)} + A_2^{(j)}$ and $A_0 + \widehat{A}_1^{(j)} + A_2^{(j)}$ are stochastic. On the other hand, it is proved (see [7, 3]) that the sequence $A_2^{(j)}$, $j \geq 0$, converges in norm to zero as σ^{2^j}, where $0 < \sigma < 1$ is given by

$$\sigma = \max\left\{|\lambda| : \det\left(A_2 + \lambda A_1 + \lambda^2 A_0 - \lambda I\right) = 0, |\lambda| < 1\right\}. \tag{5}$$

Thus, from (3), an approximation of the matrix G is given by $\left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$, for a sufficiently large value of j.

Let $\epsilon > 0$ be a fixed error bound and denote with $\boldsymbol{e}$ the k-dimensional vector having all the entries equal to 1. The stopping criterion is the stochasticity condition of $\left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$, i.e.,

$$\left(I - \left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0\right) \boldsymbol{e} < \epsilon \boldsymbol{e}. \tag{6}$$

Before describing the method presented in [1], we recall the definition of matrix sign and the definition of left invariant subspace:

Definition 1 *Let M be an $m \times m$ real matrix, with no pure imaginary eigenvalues. Let $M = S(D+N)S^{-1}$ be the Jordan decomposition of M, 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*

$$Z = sgn(M) = S\,\mathrm{Diag}(sgn(\lambda_1), sgn(\lambda_2), \ldots, sgn(\lambda_m))S^{-1},$$

where, for any complex number z with $\mathrm{re}\, z \neq 0$,

$$sgn(z) = \begin{cases} 1 & \text{if } \mathrm{re}\, z > 0 \\ -1 & \text{if } \mathrm{re}\, z < 0. \end{cases}$$

Definition 2 *Let A be an $m \times m$ real matrix. Let $\mathcal{S}$ be a k-dimensional subspace of $\boldsymbol{R}^m$ such that $A\boldsymbol{x} \in \mathcal{S}$, for any $\boldsymbol{x} \in \mathcal{S}$. Let S be an $m \times k$ matrix, whose columns are a basis of $\mathcal{S}$. Denote with A_1 a $k \times k$ matrix such that $AS = SA_1$. Then the subspace $\mathcal{S}$ is called the (closed) left invariant subspace of A if the eigenvalues of A_1 are contained in the (closed) left half-plane of the complex plane, and there is no larger subspace for which this inclusion holds.*

Let us now recall the approach proposed in [1], here adapted to the case of QBD problems. For more details we refer to [1].

Define the matrix polynomial

$$H(z) = \textstyle\sum_{i=0}^{2} H_i z^i = (1-z)^2 (tI - A_0 - A_1 t - A_2 t^2)|_{t=\frac{1+z}{1-z}} = \\ -A_0(1-z)^2 + (I - A_1)(1-z)(1+z) - A_2(1+z)^2,$$

such that $\det H(z) = 0$ has exactly $k-1$ roots with imaginary part less than zero, and a simple root at $z = 0$.

Define the matrices $\widehat{H}_0 = H_2^{-1} H_0$, $\widehat{H}_1 = H_2^{-1} H_1$ and consider the 2×2 block companion matrix

$$E = \begin{bmatrix} 0 & I \\ -\widehat{H}_0 & -\widehat{H}_1 \end{bmatrix}.$$

Let T be a $2k \times k$ matrix whose columns are a basis of the closed left invariant subspace of the matrix E, and partition the matrix T as $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$, where T_1, T_2 are $k \times k$ matrices. Then the matrix G is given by

$$G = (T_1 + T_2)(T_1 - T_2)^{-1}. \tag{7}$$

The authors observe that the closed left invariant subspace of E coincides with the left invariant subspace of the matrix

$$\widehat{E} = E - \frac{\boldsymbol{y}\boldsymbol{x}^T}{\boldsymbol{x}^T \boldsymbol{y}},$$

where

$$\boldsymbol{x}^T = \begin{bmatrix} \boldsymbol{x}_0^T \widehat{H}_1 & \boldsymbol{x}_0^T \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{0} \end{bmatrix}$$

and $\boldsymbol{x}_0$, $\boldsymbol{y}_0$ are two vectors such that $\boldsymbol{x}_0^T \widehat{H}_0 = \mathbf{0}$, $\widehat{H}_0 \boldsymbol{y}_0 = \mathbf{0}$. Now, a basis of the left invariant subspace of $\widehat{E}$, forming the columns of the matrix T, is given by k linearly independent columns of $I - Z$, where Z is the matrix sign of $\widehat{E}$. In order to compute the matrix sign Z the authors use the scaled Newton iteration (matrix sign function iteration)

$$\begin{cases} Z_0 = \widehat{E} \\ Z_{n+1} = \frac{1}{2}\left(\gamma_n Z_n + \gamma_n Z_n^{-1}\right), \ \gamma_n = |\det Z_n|^{-1/k}, \quad n \geq 0, \end{cases} \tag{8}$$

with the stopping criterion

$$||Z_{n+1} - Z_n||_1 < \epsilon ||Z_n||_1,$$

for a fixed small error bound ϵ. The above sequence quadratically converges to Z. Once Z is known, the matrix T is computed by means of the rank revealing QR decomposition of the matrix $I - Z$.

3 Theoretical and numerical comparisons

Both the algorithms consist of generating a sequence of matrices that quadratically converge to a limit, which provides the solution G. However the role of the two sequences is quite different.

In the cyclic reduction algorithm, a sequence of nonlinear matrix equations

$$X = A_0 + \widehat{A}_1^{(j)} X + A_2^{(j)} X^{2^{j+1}}, \ j \geq 0, \tag{9}$$

is generated such that its solution is the matrix G. That is, the initial problem (1) is transformed in the equivalent problems (9). Since the sequence $A_2^{(j)}$, $j \geq 0$, converges to zero, we have

$$G = \left(I - \widehat{A}_1^*\right)^{-1} A_0,$$

where $\widehat{A}_1^* = \lim_j \widehat{A}_1^{(j)}$.

In the invariant subspace approach, the authors prove that the matrix G can be recovered, by means of formula (7), after computing the matrix sign Z of $\widehat{E}$. In order to compute Z the quadratically convergent sequence (8) is generated.

Concerning the speed of convergence of the two algorithms, it is proved that the sequences $A_2^{(j)}$ and $\widehat{A}_1^{(j)}$ of cyclic reduction converge to zero and $\widehat{A}_1^*$, respectively, as σ^{2^j}, where $0 < \sigma < 1$ is defined in (5). For the sequence Z_n of (8), the authors only claim that the convergence is quadratic, and do not relate it with the blocks A_0, A_1, A_2, defining the solution G.

From (4), each step of cyclic reduction requires six $k \times k$ matrix multiplications and one matrix inversion, and thus the computational cost of one step is about $14k^3$ arithmetic operations. A matrix sign function iteration, i.e., the computation of Z_n, requires one inversion of a $2k \times 2k$ matrix, that is performed by means of LU factorization with $16k^3$ arithmetic operations.

Concerning numerical stability, the operations involved in (4) consist of multiplications and additions of nonnegative matrices and inversions of M-matrices [10], i.e., matrices having non-positive off-diagonal entries and nonnegative inverse. These computations, if the diagonal adjustment technique [6] is used, can be reduced to performing additions of positive numbers, multiplications and divisions. This avoids cancellation errors. Moreover, the matrices $\left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$ form a sequence of sub-stochastic matrices, that monotonically converges to G. If the stopping criterion (6) is satisfied, and if $\widetilde{G} = \left(I - \widehat{A}_1^{(j)}\right)^{-1} A_0$ is the corresponding approximation of G, then, from (6), we have

$$||G - \widetilde{G}||_\infty = ||(G - \widetilde{G})\boldsymbol{e}||_\infty = ||(I - \widetilde{G})\boldsymbol{e}||_\infty < \epsilon.$$

Thus condition (6) is a good stopping criterion for cyclic reduction.

An analysis of the numerical stability of the invariant subspace approach is not presented by the authors. The effect of the round-off errors generated in the computation of the matrices Z_n, in particular in the case where Z_n is close to a singular matrix, is not clear. In practice, it may happen that the sequence Z_n converges to a matrix $\widetilde{Z}$ which is not the matrix sign of $\widehat{E}$. In this case it is not clear what the error of the approximation $\widetilde{G}$ of the matrix G would be. In the extensive numerical results presented in [1], the authors report the error of the matrix $\widetilde{G}$ as $||\boldsymbol{e} - \widetilde{G}\boldsymbol{e}||_\infty$. This error does not give any information on the accuracy of the computed solution, unless $\widetilde{G}$ is sub-stochastic. Note that according to this definition of error, the identity matrix would be an approximation of the solution of any matrix equation (1) with *error zero*.

We have tested the cyclic reduction algorithm and the invariant subspace method on some examples. The cyclic reduction algorithm for QBD problems has been implemented in Fortran90 (the code can be obtained at the URL http://www.dm.unipi.it/pages/meini/public_html). For the invariant subspace method we have used the C implementation provided by the authors of [1] in the TELPACK package. This package can be taken at the URL http://www.cstp.umkc.edu/org/tn/telpack/home.html). The programs have run on an alpha workstation, with the IEEE standard double precision arithmetic. For the stopping conditions of both the algorithms we have chosen $\epsilon = 10^{-12}$. In the tables of results we have reported, for the two algorithms, the CPU time (in seconds) needed to compute an approximation $\widetilde{G}$ of the matrix G, the residual

	Cyclic Reduction			Invariant Subspace		
p	Iter.	Time	Error	Iter.	Time	Error
10^{-1}	8	0.00	$4.4 \cdot 10^{-15}$	7	0.00	$6.6 \cdot 10^{-16}$
10^{-2}	12	0.00	0.0	8	0.00	$8.0 \cdot 10^{-16}$
10^{-3}	15	0.00	0.0	8	0.00	$1.5 \cdot 10^{-15}$
10^{-4}	19	0.00	0.0	9	0.00	$2.2 \cdot 10^{-16}$
10^{-5}	22	0.00	0.0	9	0.00	0.0
10^{-6}	25	0.00	0.0	9	0.00	$6.7 \cdot 10^{-16}$
10^{-7}	29	0.00	0.0	10	0.02	$9.4 \cdot 10^{-16}$
10^{-8}	32	0.00	0.0	10	0.02	$1.3 \cdot 10^{-15}$
10^{-9}	35	0.00	0.0	10	0.00	$1.8 \cdot 10^{-15}$
10^{-10}	38	0.00	0.0	10	0.00	$1.9 \cdot 10^{-16}$

Table 1: Example 1

error of the computed approximation (as $err = ||\widetilde{G} - A_0 - A_1\widetilde{G} - A_2\widetilde{G}^2||_\infty$) and the number of iterations.

Example 1 The first problem we have tested is Example 1 of [1]. Here the blocks, having dimension 2, depend on a parameter $p > 0$, and are defined as

$$A_0 = \begin{bmatrix} 1-p & 0 \\ 0 & 0 \end{bmatrix}, \; A_1 = \begin{bmatrix} 0 & p \\ 2p & 0 \end{bmatrix}, \; A_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1-2p \end{bmatrix}.$$

The authors of [1] claim that their method "appears to be slightly faster than the logarithmic reduction algorithm of Latouche and Ramaswami" and that the accuracy of the latter algorithm "appears to deteriorate as the parameter p decreases". We have tested the same values of the parameter p of [1]. From Table 1 it follows that the performances of both the algorithms are insensitive to the parameter p. The execution times and the residual errors are negligible, for any value of p, for the cyclic reduction algorithm and for the invariant subspace method.

Example 2 The problem we now consider, already tested in [1], is the teletraffic system of [5], modeled as a continuous time QBD process, whose infinitesimal

generator is defined by the 24×24 blocks

$$(A_0')_{i,j} = \begin{cases} 192\rho_d & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (A_2')_{i,j} = \begin{cases} 192(1 - i/24) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},$$

$$(A_1')_{i,j} = \begin{cases} ar(M-i)/M & \text{if } i - j = -1 \\ ir & \text{if } i - j = 1 \\ 0 & \text{elsewhere} \end{cases}, \quad i, j = 0, 1, \ldots, 23.$$

Here a, r, M, ρ_d are parameters of the teletraffic model. We reduce the above process to a discrete-time QBD Markov chain, defined by the 24×24 blocks A_0, A_1, A_2, and we apply the algorithms for the computation of the matrix G. Tables 2 and 3 report the results obtained with the parameters tested in [1]. From Table 2 we observe that the cyclic reduction outperforms the invariant

	Cyclic Reduction			Invariant Subspace		
M	Iter.	Time	Error	Iter.	Time	Error
64	18	0.09	$7.8 \cdot 10^{-16}$	9	0.23	$3.0 \cdot 10^{-15}$
128	19	0.09	$9.9 \cdot 10^{-16}$	10	0.23	$1.6 \cdot 10^{-15}$
256	20	0.10	$1.0 \cdot 10^{-15}$	11	0.23	$4.0 \cdot 10^{-15}$
512	21	0.10	$8.0 \cdot 10^{-16}$	12	0.27	$1.4 \cdot 10^{-14}$
1024	22	0.11	$7.9 \cdot 10^{-16}$	13	0.28	$4.5 \cdot 10^{-14}$
2048	23	0.11	$8.9 \cdot 10^{-16}$	13	0.28	$1.8 \cdot 10^{-13}$
4096	24	0.12	$8.9 \cdot 10^{-16}$	14	0.30	$6.7 \cdot 10^{-13}$
8192	25	0.12	$9.1 \cdot 10^{-16}$	15	0.32	$2.9 \cdot 10^{-12}$
16384	26	0.13	$8.9 \cdot 10^{-16}$	16	0.35	$2.5 \cdot 10^{-11}$
32768	28	0.14	$7.8 \cdot 10^{-16}$	43	0.85	$2.1 \cdot 10^{-10}$
65536	33	0.16	$8.9 \cdot 10^{-16}$	*	*	*

Table 2: Example 2: $r = 1/300$, $a = 18.244$, $\rho_d = 0.280$

subspace method, both for the execution time, and for the accuracy of the result. Concerning the accuracy, we observe that cyclic reduction is insensitive to the parameter M, while the results obtained with the invariant subspace approach deteriorate, as M grows. In the case $M = 65536$ the TELPACK program has an abnormal termination, with the segmentation fault signal. By increasing to $\epsilon = 10^{-8}$ the threshold value for the stopping condition, the program terminates after 133 iterations, in 258 seconds, with the residual error $7.0 \cdot 10^{-7}$.

Analogous results are obtained in Table 3, where r, a and M are fixed. Also in this case the cyclic reduction outperforms the invariant subspace approach.

Indeed, the accuracy of the results of the latter method deteriorates, as ρ_d grows, while cyclic reduction seems more robust. Moreover, in the case $\rho_d = 0.29568$, which is close to $\rho_d = 0.296$, that yields an unstable queuing system, the TELPACK program has an abnormal termination, with the segmentation fault signal. By increasing to $\epsilon = 10^{-8}$ the threshold value for the stopping condition, the program terminates after 83 iterations, in 1.63 seconds, with the residual error $3.6 \cdot 10^{-8}$.

	Cyclic Reduction			Invariant Subspace		
ρ_d	Iter.	Time	Error	Iter.	Time	Error
0.01	5	0.03	$8.9 \cdot 10^{-16}$	10	0.25	$5.3 \cdot 10^{-16}$
0.025	6	0.03	$9.5 \cdot 10^{-16}$	10	0.23	$8.4 \cdot 10^{-16}$
0.05	9	0.05	$8.9 \cdot 10^{-16}$	10	0.23	$3.0 \cdot 10^{-15}$
0.075	11	0.06	$1.2 \cdot 10^{-15}$	10	0.23	$1.7 \cdot 10^{-15}$
0.1	13	0.07	$9.0 \cdot 10^{-16}$	10	0.23	$1.3 \cdot 10^{-15}$
0.12	13	0.06	$1.0 \cdot 10^{-15}$	10	0.23	$6.5 \cdot 10^{-16}$
0.14	14	0.07	$7.8 \cdot 10^{-16}$	10	0.23	$1.5 \cdot 10^{-15}$
0.16	15	0.07	$9.0 \cdot 10^{-16}$	9	0.20	$9.0 \cdot 10^{-16}$
0.18	15	0.07	$1.0 \cdot 10^{-15}$	9	0.22	$8.5 \cdot 10^{-16}$
0.2	16	0.08	$8.9 \cdot 10^{-16}$	9	0.20	$1.0 \cdot 10^{-15}$
0.22	17	0.08	$8.1 \cdot 10^{-16}$	9	0.22	$3.0 \cdot 10^{-15}$
0.24	17	0.08	$1.0 \cdot 10^{-15}$	10	0.25	$2.4 \cdot 10^{-15}$
0.26	18	0.09	$8.9 \cdot 10^{-16}$	10	0.23	$4.0 \cdot 10^{-15}$
0.28	19	0.09	$8.1 \cdot 10^{-16}$	12	0.27	$2.8 \cdot 10^{-14}$
0.29	21	0.10	$6.7 \cdot 10^{-16}$	13	0.28	$1.0 \cdot 10^{-13}$
0.29568	30	0.12	$6.8 \cdot 10^{-16}$	*	*	*

Table 3: Example 2: $r = 1/100$, $a = 18.244$, $M = 512$

Example 3 In this example we construct a QBD problem defined by the $k \times k$ blocks $A_0 = R + \delta I$, $A_1 = A_2 = R$, where R is a matrix having null diagonal entries and constant off-diagonal entries, and $0 < \delta < 1$. We observe that, in this case, the rate $\rho = \boldsymbol{a}^T(A_1 + 2A_2)\boldsymbol{e}$, where $\boldsymbol{a}^T(A_0 + A_1 + A_2) = \boldsymbol{a}^T$, $\boldsymbol{a}^T\boldsymbol{e} = 1$, is exactly $\rho = 1 - \delta$. Thus, as δ approaches zero, the Markov chain becomes more unstable. Here we test the two algorithms for different values of the parameter δ. Tables 4, 5, 6 report the results obtained with size $k = 16$, $k = 32$, $k = 64$. In the last column of each table we also report the closeness to stochasticity of the matrix $\widetilde{G}$ obtained with the invariant subspace approach, i.e., $||\boldsymbol{e} - \widetilde{G}\boldsymbol{e}||_\infty$.

	Cyclic Reduction			Invariant Subspace			
δ	Iter.	Time	Error	Iter.	Time	Error	Stoc.
10^{-1}	7	0.01	$4.8 \cdot 10^{-16}$	8	0.07	$6.1 \cdot 10^{-15}$	$1.3 \cdot 10^{-15}$
10^{-2}	10	0.02	$5.6 \cdot 10^{-16}$	11	0.08	$8.1 \cdot 10^{-14}$	$4.7 \cdot 10^{-15}$
10^{-3}	13	0.02	$5.1 \cdot 10^{-16}$	13	0.10	$8.0 \cdot 10^{-12}$	$2.9 \cdot 10^{-15}$
10^{-4}	16	0.03	$6.5 \cdot 10^{-16}$	16	0.12	$6.7 \cdot 10^{-10}$	$1.3 \cdot 10^{-15}$
10^{-5}	20	0.03	$5.3 \cdot 10^{-16}$	18	0.12	$5.9 \cdot 10^{-8}$	$3.1 \cdot 10^{-15}$
10^{-6}	23	0.04	$7.1 \cdot 10^{-16}$	25	0.15	$4.4 \cdot 10^{-6}$	$3.2 \cdot 10^{-15}$
10^{-7}	26	0.05	$6.7 \cdot 10^{-16}$	69	0.45	$4.3 \cdot 10^{-4}$	$1.8 \cdot 10^{-15}$
10^{-8}	29	0.05	$6.3 \cdot 10^{-16}$	25	0.17	$1.4 \cdot 10^{-2}$	$1.3 \cdot 10^{-15}$

Table 4: Example 3: $k = 16$

	Cyclic Reduction			Invariant Subspace			
δ	Iter.	Time	Error	Iter.	Time	Error	Stoc.
10^{-1}	7	0.08	$9.9 \cdot 10^{-16}$	8	0.45	$7.8 \cdot 10^{-15}$	$3.1 \cdot 10^{-15}$
10^{-2}	10	0.11	$1.2 \cdot 10^{-15}$	11	0.58	$1.1 \cdot 10^{-13}$	$7.8 \cdot 10^{-15}$
10^{-3}	13	0.14	$9.8 \cdot 10^{-16}$	14	0.72	$9.4 \cdot 10^{-12}$	$4.2 \cdot 10^{-15}$
10^{-4}	16	0.19	$8.9 \cdot 10^{-16}$	17	0.85	$1.0 \cdot 10^{-9}$	$4.0 \cdot 10^{-15}$
10^{-5}	20	0.22	$8.5 \cdot 10^{-16}$	24	1.17	$6.6 \cdot 10^{-8}$	$1.7 \cdot 10^{-14}$
10^{-6}	23	0.27	$1.2 \cdot 10^{-15}$	93	4.43	$6.3 \cdot 10^{-6}$	$8.2 \cdot 10^{-15}$
10^{-7}	26	0.30	$1.3 \cdot 10^{-15}$	31	1.52	$8.1 \cdot 10^{-4}$	$2.0 \cdot 10^{-15}$
10^{-8}	29	0.33	$6.2 \cdot 10^{-16}$	55	2.65	$4.9 \cdot 10^{-2}$	$2.4 \cdot 10^{-15}$

Table 5: Example 3: $k = 32$

	Cyclic Reduction			Invariant Subspace			
δ	Iter.	Time	Error	Iter.	Time	Error	Stoc.
10^{-1}	7	0.76	$2.8 \cdot 10^{-15}$	8	3.87	$2.5 \cdot 10^{-14}$	$2.4 \cdot 10^{-14}$
10^{-2}	10	1.07	$1.9 \cdot 10^{-15}$	11	5.05	$2.6 \cdot 10^{-13}$	$2.1 \cdot 10^{-14}$
10^{-3}	13	1.38	$2.5 \cdot 10^{-15}$	14	6.30	$1.7 \cdot 10^{-11}$	$3.5 \cdot 10^{-14}$
10^{-4}	16	1.69	$2.0 \cdot 10^{-15}$	17	7.50	$2.0 \cdot 10^{-8}$	$1.1 \cdot 10^{-14}$
10^{-5}	20	2.10	$2.4 \cdot 10^{-15}$	20	8.70	$1.8 \cdot 10^{-7}$	$7.1 \cdot 10^{-15}$
10^{-6}	23	2.46	$2.3 \cdot 10^{-15}$	59	24.90	$2.3 \cdot 10^{-5}$	$3.5 \cdot 10^{-14}$
10^{-7}	26	2.79	$3.3 \cdot 10^{-15}$	43	18.30	$2.1 \cdot 10^{-3}$	$4.4 \cdot 10^{-11}$
10^{-8}	29	3.04	$1.7 \cdot 10^{-15}$	78	32.70	$1.8 \cdot 10^{-2}$	$5.9 \cdot 10^{-15}$

Table 6: Example 3: $k = 64$

This value is reported by the TELPACK package as an estimate of the accuracy of the output. It is worth pointing out that, even though this value is small, the corresponding approximation error may be large. In fact, for small values of δ, the invariant subspace algorithm delivers an output matrix that, though stochastic, is a very poor approximation of the solution of (1). On the other hand, as shown by tables 4–6, the cyclic reduction algorithm shows its robustness and effectiveness for any value of δ.

References

[1] N. Akar and K. Sohraby. An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. *Commun. Statist. Stochastic Models*, 13:381–416, 1997.

[2] D. Bini and B. Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 21–38. Kluwer Academic Publisher, 1995.

[3] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:906–926, 1996.

[4] D. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 15:57–74, 1997.

[5] J. N. Daigle and D. M. Lucantoni. Queueing systems having phase-dependent arrival and service rates. In W. J. Stewart, editor, *Numerical solution of Markov Chains*, pages 161–202. Marcel Dekker, New York, 1991.

[6] W. K. Grassman, M. I. Taksar, and D. P. Heyman. Regenerative analysis and steady state distribution for Markov chains. *Oper. Res.*, 33:1107–1116, 1985.

[7] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Probability*, 30:650–674, 1993.

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

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

[10] R. S. Varga. *Matrix Iterative Analysis.* Prentice Hall, New Jersey, 1962.

5 N-Burst Processes as Models of Self-similar Traffic in Telecommunication Systems – I. Introduction and LAQT Background*

Lester Lipsky† **& Hans-Peter Schwefel**‡

Abstract

A short history of what we call "Linear Algebraic Queueing Theory (LAQT)" is presented, including the basic formulas for matrix representations of Phase (or 'Matrix Exponential') Distributions. Semi-Markov (or Markov Renewal) Processes are then described in this formulation, with particular emphasis on 'Markov Modulated Poisson Processes (MMPP)'. The MMPP process is then modified to represent ON-OFF models with non-exponential holding times (the 1-Burst model). A 'Truncated Power-Tail' (TPT) Distribution with a Phase representation is then described. It is the combination of ON-OFF Processes with TPT ON-times that allows telecommunication systems with self-similar and long-range dependent traffic to be modeled analytically.

5.1 Introduction

Research of telecommunications traffic over the last decade has led to the realization that standard models of performance have to be modified or expanded to include long-range autocorrelation of interarrival times of data packets, and large fluctuations in the number of arrivals for any time intervals, the so-called *self-similarity property*. Analytic models can use *Markov Arrival Processes* (MAP) to include autocorrelation, but long-range correlation and self-similarity require that *Power-tail* distributions be included in some way. Our goal in this paper is to introduce Power-Tail distributions into the matrix-algebraic solution techniques. Therefore, a considerable amount of background material is included for those who are not very familiar with, what we call, *Linear Algebraic Queueing Theory* (LAQT).

After a short history of how non-exponential distributions have come to fit into Markov chain theory, and then some basic definitions, we describe *Matrix Exponential* (ME) distributions and their properties. In Section 5.3.3 we give the solution to M/G/1 queues

* Research (partly) funded by Deutsche Telekom AG.

† Department of Computer Science and Engineering, University of Connecticut, Storrs, U.S.A.
E-mail: lester@brc.uconn.edu

‡ Institut für Informatik, Technische Universität München, Germany
E-mail: schwefel@in.tum.de

in terms of ME functions. In Section 5.4 we introduce *Semi-Markov* (SM) processes in terms of the generators of ME distributions, showing how they can be used to compute the autocorrelation coefficients lag-k, and how to solve SM/M/1 queues, the basis for our model of the performance of telecommunication servers (routers or packet switches). We then present the most-used type of SM process, Markov Modulated Poisson Processes (MMPP), and how to embed non-exponential distributions into them. We discuss the simplest non-trivial example of a 1-Burst, ON-OFF model.

Section 5.5 contains a brief discussion of how to build matrix representations of a particular class of *Truncated Power Tail* (TPT) distributions and their properties. Finally, we combine the TPT's with our form of MMPP and give some preliminary results for the 1-Burst model. The companion paper, [Lipsky & Schwefel, 1999] (hereafter referred to as *Part II*), will go into detail of how to extend the model to various types of N-Burst models which are better suited for analyzing busy telecommunication systems.

5.2 A Short History of LAQT

K. Erlang (1900's): *Erlangian Distributions and Method of Stages*;
Erlang started modern queueing theory, and even then tried to build non-exponential service times by having identical exponential stages in tandem.

D. R. Cox (1950's): *Coxian Servers; Completeness;*
Looked at Tandem stages with unequal service times, and finally showed that if *complex service times*, and jump-forward probabilities are allowed, then any distribution can be approximated arbitrarily closely.

D. G. Kendall (1950's): *Coxian Servers Have Rational Laplace Transforms (RLT);*
The set of RLT distributions (also called *Kendall distributions*) is *complete* in the above sense, and therefore transform methods are useful (analytic expressions for inverse transforms are known only for RLT functions).

V. Wallace (1960's): *Quasi-Birth-Death (QBD) Processes and Matrix-Geometric Solutions;*
Examined Markov chains where the generator is *block tri-diagonal*, and solved them in the same way that one solves birth-death processes. He showed that infinite, repeating QBD processes have a matrix geometric solution.

M. Neuts (1975): *Phase Distributions;*
Showed that *sub-stochastic processes* (Markov chains with an *absorbing* state) whose states can have unequal service times, have general service times that can be made to approach any distribution with the increase of number of states (now called *phases* because they are not necessarily physical), while strictly adhering to Markovian properties (real service times, etc).

M. Neuts, L. Lipsky, and others (1980's): *Explicit Matrix Algebraic Solutions of M/G/1, GI/M/C, GI/G/1//N, M/G/C Queues, etc.;*
Explicit matrix analytic solutions of various steady-state queueing systems were derived formally using Phase (or more generally, Matrix Exponential) distributions.

L. Lipsky (1990's): *Linear Algebraic Queueing Theory (LAQT);*
Matrices are considered as linear operators which change the state of a system in correspondence with actual dynamic changes of a system, with the potential for solving transient, as well as steady-state systems.

5.3 Matrix Exponential (ME) Distributions

The material in this section is described fully in [Lipsky, 1992], Chapters III and IV. After stating some elementary definitions, we derive the properties of ME distributions based on a *token's* travel through a *black box*. They are then used to state some properties of M/G/1 queues.

5.3.1 Distribution Functions – Definitions

- Probability Distribution Function (PDF):

$$F(x) := Pr(X \leq x) = \text{Probability that service will be completed by } x$$

- Reliability Function and probability density function (pdf):

$$R(x) := 1 - F(x); \qquad f(x) := \frac{d}{dx}F(x) = -\frac{d}{dx}R(x)$$

- Expected Values (Averages):

$$\mathrm{E}(X) := \int_0^\infty x\, f(x)\, dx \quad \text{(Mean)}; \qquad \mathrm{E}(X^\ell) := \int_0^\infty x^\ell\, f(x)\, dx \quad (\ell^{th}\,\text{Moment})$$

- Variance:

$$\sigma^2 := \mathrm{E}[(X - \mathrm{E}(X))^2] = \mathrm{E}(X^2) - [\mathrm{E}(X)]^2$$

- Standard Deviation and Coefficient of Variation:

$$\sigma := \sqrt{\sigma^2}; \qquad C^2 := \sigma^2 / \mathrm{E}\,(x)^2$$

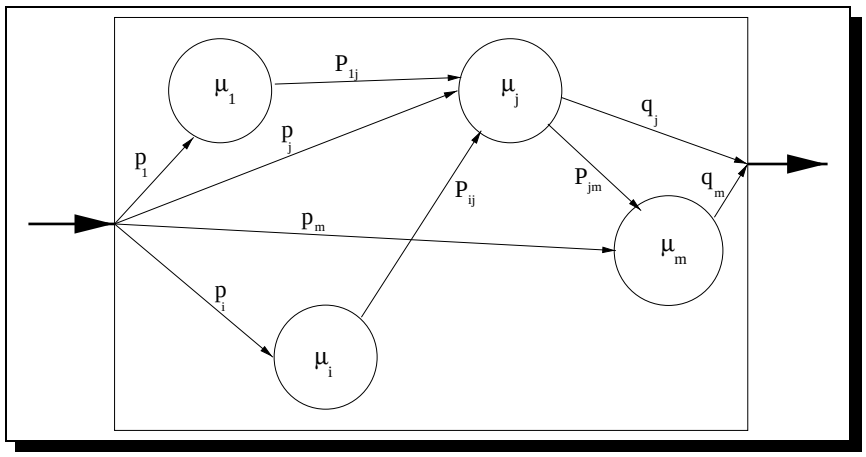


Figure 5.1: *A Black Box containing m phases:* a *token* enters and goes to phase i with probability p_i. After spending an exponential time with mean $1/\mu_i$, it goes to phase j with probability P_{ij}, or leaves with probability q_i. The box is black because an outside observer cannot "see" where the token is once it enters.

5.3.2 Properties of ME Distributions

Consider Figure 5.1. There are m *phases* (also, for now, *stages* or *states*) in the box. Let a customer or *token* enter the box and go to phase i with probability p_i. Call $\mathbf{p}$ the *entrance vector*, where

$$\mathbf{p} := [p_1, p_2, \ldots, p_m], \quad \text{and} \quad \boldsymbol{\epsilon} := [1, 1, \ldots, 1].$$

Then $\mathbf{p}\boldsymbol{\epsilon}' = \sum_{j=1}^{m} p_j = 1$, where $\boldsymbol{\epsilon}'$ is the transpose of $\boldsymbol{\epsilon}$, i.e., a column vector of all 1's. P_{ij} is the probability that the *token* will go to phase j, given that it has completed service at i. Then $(\mathbf{P})_{ij} := P_{ij}$ is a *sub-stochastic* matrix. That is, $(\mathbf{P}\boldsymbol{\epsilon}')_i = \sum_{j=1}^{m} P_{ij} < 1$ for at least one i (i.e., the token can get out of the box). Let q_i be the probability that the token can leave the box directly from phase i. Then

$$\mathbf{q}' = (\mathbf{I} - \mathbf{P})\boldsymbol{\epsilon}' \tag{5.1}$$

Let $\mathbf{M}$ be a diagonal matrix with $(\mathbf{M})_{ii} = \mu_i$, the (exponential) service rate of phase i. Clearly, $(\mathbf{M}\boldsymbol{\epsilon}')_i = \mu_i$. Next define

$$\tau_i := \text{Mean time for token to leave box, given that it started at phase } i.$$

The token spends a mean time of $1/\mu_i$ at phase i, and then either leaves the box or goes to phase j with probability P_{ij}, and then takes a time τ_j to eventually leave. That is,

$$\tau_i = \frac{1}{\mu_i} + \sum_{j=1}^{m} P_{ij}\tau_j.$$

This can be written in matrix form as:

$$\boldsymbol{\tau}' = \mathbf{M}^{-1}\boldsymbol{\epsilon}' + \mathbf{P}\boldsymbol{\tau}'.$$

Solve algebraically for $\boldsymbol{\tau}'$ to get

$$\boldsymbol{\tau}' = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{M}^{-1}\boldsymbol{\epsilon}'.$$

Let

$$\mathbf{B} := \mathbf{M}(\mathbf{I} - \mathbf{P}) \quad \text{and} \quad \mathbf{V} = \mathbf{B}^{-1}. \tag{5.2}$$

[For future reference, (5.2) and (5.1) imply that $\mathbf{Mq}' = \mathbf{B}\boldsymbol{\epsilon}$.] Then $\boldsymbol{\tau}' = \mathbf{V}\boldsymbol{\epsilon}'$. Now, p_i is the probability that the token, upon entering the box, goes directly to phase i. Let X be the random variable representing the time the token spends in the box. Therefore, the mean time the token spends in the box is given by

$$\mathrm{E}(X) = \sum_{i=1}^{m} p_i \tau_i = \mathbf{p}\boldsymbol{\tau}' = \mathbf{p}\,\mathbf{V}\,\boldsymbol{\epsilon}' =: \Psi[\mathbf{V}].$$

More generally, define

$$r_i(t) := Pr(\text{`token is at phase } i \text{ at time } t\text{'}).$$

with probability vector, $\mathbf{r}(t) := [r_1(t),\, r_2(t),\, \cdots,\, r_m(t)\,]$. Then

$$\mathbf{r}(t)\boldsymbol{\epsilon}' = \textstyle\sum_i r_i(t) = Pr(\text{`token is still in the box at time } t\text{'}).$$

Consider the system a small time δ later. The probability that the token will be at phase i at time $t+\delta$ is equal to the probability, $(1-\mu_i\delta)$, that it was there at time t and nothing happened in the interval of length δ, plus the probability, $(\mu_j\,\delta\,P_{ji})$, that it was at some other phase, j, left there in the interval δ, and then went to i, plus other multiple events of order δ^2. The equation for this is:

$$r_i(t+\delta) = r_i(t)(1-\mu_i\delta) + \sum_j r_j(t)\mu_j\delta P_{ji} + O(\delta^2).$$

Rearrange terms, divide by δ, let $\delta \to 0$, then

$$\frac{d\mathbf{r}(t)}{dt} = -\mathbf{r}(t)\,\mathbf{B}.$$

The solution of this matrix differential equation is

$$\mathbf{r}(t) = \mathbf{r}(0)\exp(-t\mathbf{B})$$

where the matrix, $\exp(\mathbf{X})$, is defined by its Taylor series expansion ($e^x = 1 + x + x^2/2! + x^3/3! + \cdots$). If the token enters the box at time $t = 0$, then $\mathbf{r}(0) = \mathbf{p}$, and the reliability function for the time spent in the box is given by

$$R(t) = Pr(X > t) = \mathbf{p}\,\exp(-t\mathbf{B})\,\boldsymbol{\epsilon}' = \Psi[\,\exp(-t\mathbf{B})]. \tag{5.3}$$

So, the vector-matrix pair, $<\,\mathbf{p},\,\mathbf{B}\,>$, *generates* a *Matrix Exponential* (ME) representation of $R(t)$. It can be shown that as long as all the eigenvalues of $\mathbf{B}$ have positive real parts, the matrix function, $\exp(-t\mathbf{B})$ has the same properties as its scalar counterpart. Some of these properties are:

$$f(t) = -\frac{d}{dt}R(t) = \Psi[\exp(-t\mathbf{B})\,\mathbf{B}]$$

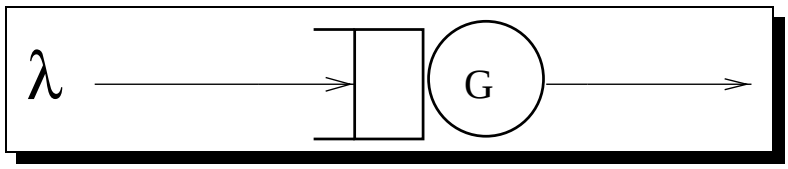


Figure 5.2: *Open M/G/1 system:* G is a black box, and number of customers waiting in the queue can become unboundedly large.

and

$$\mathrm{E}(X^\ell) = \mathbf{p}\left[\int_0^\infty t^\ell \exp(-t\mathbf{B})\,\mathbf{B}\,dt\right]\boldsymbol{\epsilon}' = \ell!\,\Psi[\mathbf{V}^\ell]. \tag{5.4}$$

The Laplace Transform also adheres to this, namely

$$F^*(s) := \int_0^\infty e^{-st} f(t)\,dt = \Psi[(\mathbf{I} + s\mathbf{V})^{-1}]. \tag{5.5}$$

It is easy to show that if $\mathrm{Dim}(\mathbf{V})$ is finite, then $F^*(s)$ is a ratio of polynomials in s. That is, $F^*(\cdot)$ is the *Rational Laplace Transform* (RLT) of $f(\cdot)$. We mention here that $f(\cdot)$ is RLT and *finite* ME iff it can be written as a sum of terms of the form

$$\{t^k\, e^{-\mu t} \mid k \geq 0;\ \Re(\mu) > 0\}\ .$$

Erlangian distributions (but not fractional Gamma functions), hyper-exponential distributions, hypo-exponential (or generalized Erlangian) distributions, and linear combinations of them, are all RLT and special examples of ME distributions. Coxian servers form an equivalent class, while Phase distributions form a proper subset of ME distributions.

5.3.3 Steady-state M/G/1 Queue

If one considers a black box to be the general server in an M/G/1 queue (see Figure 5.2) then the resultant balance equations can be written down and solved explicitly. The solutions match the Pollaczek-Khinchin formulas, and even yield the system time distributions. We summarize the solutions here. Let λ be the arrival rate of the customers, and

$$\mathbf{A} := \mathbf{I} + \frac{1}{\lambda}\mathbf{B} - \boldsymbol{\epsilon}'\mathbf{p}, \quad \text{with} \quad \mathbf{U} := \mathbf{A}^{-1}.$$

The steady-state solution vector, $\mathbf{r}(n)$, is given by

$$\mathbf{r}(n) = (1-\rho)\mathbf{p}\,\mathbf{U}^n$$

where its i^{th} component, $r_i(n)$, is the probability that a random observer will find n customers at the general queue, and the token for the active customer is at phase i. The utilization is $\rho = \lambda\Psi[\mathbf{V}]$. The associated scalar probability is given by

$$r(n) = \mathbf{r}(n)\boldsymbol{\epsilon}' = (1-\rho)\Psi[\mathbf{U}^n].$$

In comparing with the simple M/M/1 queue, we note that the exponential distribution has a 1-dimensional representation, so $\mathbf{B} \to \mu$ (a scalar), $\mathbf{p} \to 1$, $\boldsymbol{\epsilon}' \to 1$, and $\mathbf{U} \to \lambda/\mu = \rho$.

5.4 Semi-Markov Processes (SMP) Markov Renewal (MRP), Markov Arrival (MAP)

Much of the material in this section is described in detail in [Fiorini et al., 1995]. If the general server in the previous section is so busy that the probability it will be idle goes to zero, then the times between departures are distributed according to $< \mathbf{p}, \mathbf{B} >$. After each departure, the next token starts in vector state $\mathbf{p}$. Therefore, the inter-departure times are independent (all start with the same probability vector) and identically distributed (same $\mathbf{B}$). Thus when $\rho \geq 1$, the queue becomes unboundedly large, but departures from the box constitute a renewal process to whatever is downstream.

Suppose instead that each departure process starts with a different initial vector depending on the previous departure. Then the inter-departure times may be differently distributed, *and* correlated. Let

$$\mathbf{p_i} = [(\mathbf{p_i})_1, (\mathbf{p_i})_2, \cdots, , (\mathbf{p_i})_m]$$

with $\mathbf{p_i}\boldsymbol{\epsilon}' = 1$ for $1 \leq i \leq m$, be the starting vector for an inter-departure time, given that the previous event started in state i. Then

$$\boldsymbol{\mathcal{Y}} := \begin{bmatrix} (\mathbf{p_1})_1 & (\mathbf{p_1})_2 & \cdots & (\mathbf{p_1})_m \\ (\mathbf{p_2})_1 & (\mathbf{p_2})_2 & \cdots & (\mathbf{p_2})_m \\ \cdots & \cdots & \cdots & \cdots \\ (\mathbf{p_m})_1 & (\mathbf{p_m})_2 & \cdots & (\mathbf{p_m})_m \end{bmatrix}.$$

Each row is an m-dimensional starting vector, and thus, $\boldsymbol{\mathcal{Y}}$ is a Markov matrix, since $\boldsymbol{\mathcal{Y}}\boldsymbol{\varepsilon}' = \boldsymbol{\varepsilon}'$.

Next let $\boldsymbol{\mathcal{B}}$ be the generator of the inter-departure times. (Here we assume that we have the same $\boldsymbol{\mathcal{B}}$ for all the $\mathbf{p_i}$'s. A more general assumption is unnecessary.) Then

$$[\exp(-t\boldsymbol{\mathcal{B}})\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{Y}}]_{ij}\delta$$

is the probability that a departure occurred in some interval, δ, around t *and* the next epoch will start in phase j, given that the token was in phase i at time $t = 0$. From this it can be seen that

$$\boldsymbol{\mathcal{L}} := \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{Y}}$$

is the infinitesimal generator of the next departure. First note that

$$\boldsymbol{\mathcal{L}}\boldsymbol{\varepsilon}' = \boldsymbol{\mathcal{B}}\boldsymbol{\varepsilon}'.$$

Then it can be shown that $\delta[\boldsymbol{\mathcal{L}}]_{ij}$ is the probability that a departure will occur in the next time-interval, δ, and the next epoch will start in phase j, given that the token is in state i.

5.4.1 Joint Distributions

From the definitions of $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{L}}$, we can write down the joint probabilities of successive inter-departure times. Let $\{X_\kappa | \kappa = 0, 1, \cdots\}$ be the random variables denoting the inter-departure times. Let $\boldsymbol{\wp}_0$ be the vector state of the system at time $t = 0$. Then,

$$f_{X_0 \cdots X_\ell}(t_0, \cdots, t_\ell) = \boldsymbol{\wp}_0[\exp(-t_0\,\boldsymbol{\mathcal{B}})\,\boldsymbol{\mathcal{L}}][\exp(-t_1\,\boldsymbol{\mathcal{B}})\,\boldsymbol{\mathcal{L}}] \cdots [\exp(-t_\ell\boldsymbol{\mathcal{B}})\boldsymbol{\mathcal{L}}]\boldsymbol{\varepsilon}'.$$

The joint density function for any two interarrival times, say κ and $\kappa + k$ can be found by integrating over all other variables, getting

$$f_{X_\kappa\, X_{\kappa+k}}(x,t) = \boldsymbol{\wp}_0 \boldsymbol{\mathcal{Y}}^\kappa [\exp(-x\boldsymbol{\mathcal{B}})\boldsymbol{\mathcal{L}}]\, \boldsymbol{\mathcal{Y}}^k [\exp(-t\boldsymbol{\mathcal{B}})\boldsymbol{\mathcal{L}}]\boldsymbol{\varepsilon}'. \tag{5.6}$$

This formula shows that if $\boldsymbol{\wp}_0$ is given, then the κ^{th} arrival begins with vector state $\boldsymbol{\wp}_\kappa := \boldsymbol{\wp}_0 \boldsymbol{\mathcal{Y}}^\kappa$. We also see that the joint distribution is *not* the scalar product of two separate functions. Instead, the representations of X_κ and $X_{\kappa+k}$ are coupled together by the matrix, $\boldsymbol{\mathcal{Y}}^k$. Only if $\boldsymbol{\mathcal{Y}}$ is of rank 1 (i.e., $\boldsymbol{\mathcal{Y}} = \boldsymbol{\varepsilon}'\boldsymbol{\wp}$) can $f_{X_\kappa\, X_{\kappa+k}}(x,t)$ be written as $f_1(x)f_2(t)$, in which case the process reduces to a pure renewal process.

5.4.2 Auto-Correlation

From probability theory, the covariance of two random variables is given by

$$\mathrm{Cov}(X,Y) = \int_A [x - \mathrm{E}(X)][y - \mathrm{E}(Y)] f_{XY}(x,y)dxdy = \mathrm{E}(XY) - \mathrm{E}(X)\mathrm{E}(Y)$$

The *normalized* correlation coefficient is defined as

$$\varrho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\sigma_X^2\, \sigma_Y^2}}$$

and has the property, $-1 \le \varrho(X,Y) \le 1$. Let $X = X_\kappa$, and $Y = X_{\kappa+k}$, and use (5.6) to get the *auto-covariance lag-k*

$$\mathrm{Cov}(X_\kappa\, X_{\kappa+k}) = \boldsymbol{\wp}_\kappa [\boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{Y}}^k\boldsymbol{\mathcal{V}}]\boldsymbol{\varepsilon}' - [\boldsymbol{\wp}_\kappa \boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'][\boldsymbol{\wp}_\kappa \boldsymbol{\mathcal{Y}}^k\, \boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'],$$

where $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{B}}^{-1}$ analog to formula (5.2). In general, $\boldsymbol{\wp}_0$ (and thus $\boldsymbol{\wp}_\kappa$) is not known, but if a system has been running for a *long time* then κ is large, and $\boldsymbol{\wp}_\kappa \to \boldsymbol{\wp}$, where $\boldsymbol{\wp}$ satisfies the eigenvector equation $\boldsymbol{\wp}\boldsymbol{\mathcal{Y}} = \boldsymbol{\wp}$. Then the covariance becomes independent of κ, and the *auto-correlation function lag-k*, $\varrho(k)$, is given by:

$$\varrho(k) := \frac{\mathrm{Cov}(X, X_{+k})}{\sigma_X^2} = \frac{\boldsymbol{\wp}\boldsymbol{\mathcal{V}}[\boldsymbol{\mathcal{Y}}^k - \boldsymbol{\varepsilon}'\boldsymbol{\wp}]\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'}{2\boldsymbol{\wp}\boldsymbol{\mathcal{V}}^2\boldsymbol{\varepsilon}' - (\boldsymbol{\wp}\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}')^2}\,. \tag{5.7}$$

It is easy to show that if m is finite, then $\varrho(k)$ should go to 0 geometrically as ϕ^k, where ϕ is the second largest eigenvalue of $\boldsymbol{\mathcal{Y}}$ in magnitude ($|\phi| < 1$, since the largest eigenvalue of $\boldsymbol{\mathcal{Y}}$ is 1). The significance here is to note that many measurements of telecommunications traffic have indicated that the covariance of packet interarrival times drops off much more slowly, as $1/k^{\alpha-1}$. This tells us that any matrix representation that hopes to show such behavior must be infinite dimensional, or at least, m must be very large. We will present such a function in Section 5.5.2.

5.4.3 SM/M/1 Queues

The discussions of the previous sections are useful for more than computing correlation. SM processes are used to model arrival patterns to exponential and even more general servers. See [Fiorini, 1997], [Schwefel, 1997], and [Neuts, 1981]. Neuts has shown

that a matrix geometric solution for the steady state can be found for SM/M/1 (and numerous other) queues by finding an appropriate matrix, $\boldsymbol{\mathcal{R}}$, that satisfies the matrix-quadratic equation

$$\nu\boldsymbol{\mathcal{R}}^2 - \boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{B}} + \nu\boldsymbol{\mathcal{I}}) + \boldsymbol{\mathcal{L}} = \mathbf{0}.$$

The matrices, $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{B}}$ were discussed in the previous section, and ν is the service rate of the exponential server downstream from the semi-Markov generator/box. The steady-state solution is given by

$$r(n) = \boldsymbol{\pi}\,(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{R}})\boldsymbol{\mathcal{R}}^n\,\boldsymbol{\varepsilon}',$$

where $\wp\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'$ is the mean interarrival time, and

$$\boldsymbol{\pi} := \frac{\wp\boldsymbol{\mathcal{V}}}{\wp\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'}.$$

Note that (since $\boldsymbol{\mathcal{Y}}\boldsymbol{\varepsilon}' = \boldsymbol{\varepsilon}'$)

$$\boldsymbol{\pi}\boldsymbol{\mathcal{L}}\boldsymbol{\varepsilon}' = \frac{\wp\boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{L}}\boldsymbol{\varepsilon}'}{\wp\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'} = \frac{\wp\boldsymbol{\mathcal{Y}}\boldsymbol{\varepsilon}'}{\wp\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'} = \frac{1}{\wp\boldsymbol{\mathcal{V}}\boldsymbol{\varepsilon}'}.$$

Part II will discuss this in detail.

5.4.4 Markov Modulated Poisson Process (MMPP)

The most widely used semi-Markov process is the MMPP. In particular, it has been used to model voice traffic for many years. It also seems to be quite appropriate for describing so-called *ON-OFF* models. First we give a general description of MMPP's, and then we describe the particular class of interest to us.

Let $\boldsymbol{\mathcal{P}}$ be a transition matrix for some Markov chain, and let $(\mu_i)^{-1}$ be the mean time a token spends in state i. Note that here we are dealing with a token which can never leave its box ($\boldsymbol{\mathcal{P}}\boldsymbol{\varepsilon}' = \boldsymbol{\varepsilon}'$). Then $\boldsymbol{\mathcal{M}} = \mathrm{Diag}(\mu_1, \mu_2, \cdots, \mu_m)$ and

$$\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{P}}) \quad \text{with} \quad \boldsymbol{\mathcal{Q}}\boldsymbol{\varepsilon}' = \mathbf{o}'.$$

is the generator of the token's travels in the box.

Suppose that while the token is at state i, *cells* (packets) are emitted from the box at Poisson rate λ_i. Then it can be shown this constitutes a MMPP where

$$\boldsymbol{\mathcal{L}} = \mathrm{Diag}(\lambda_1, \lambda_2, \cdots, \lambda_m), \quad \text{and} \quad \boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{Q}} + \boldsymbol{\mathcal{L}}. \tag{5.8}$$

It turns out that $\boldsymbol{\pi}$ with $\boldsymbol{\pi}\boldsymbol{\mathcal{Q}} = \mathbf{o}$ is the steady-state vector for the token's travels in the box. Then, with $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{L}}$,

$$\wp = \frac{\boldsymbol{\pi}\boldsymbol{\mathcal{L}}}{\boldsymbol{\pi}\boldsymbol{\mathcal{L}}\boldsymbol{\varepsilon}'} \tag{5.9}$$

is the unit eigenvector satisfying $\wp\boldsymbol{\mathcal{Y}} = \wp$.

In summary, cells leave the system with exponential interdeparture times (Poisson Process), but the departure rate is modulated (changed) according to the Markov chain describing the token's travels. Hence the name, MMPP.

5.4.5 'Generalized' MMPP

The assumption that the length of time for a fixed cell rate be exponentially distributed is restrictive. We can let the token spend a non-exponential amount of time emitting cells at rate λ_i in the following way. Let $< \mathbf{p_i}, \mathbf{B_i} >$ (together with the associated matrices, $\mathbf{P_i}, \mathbf{M_i}, \mathbf{V_i}, \mathbf{q_i'}$) be the m_i-dimensional generator of the time the token spends in stage i before moving to the next stage with probability $(\boldsymbol{P})_{ij}$.

Before going on, a brief discussion of the notation used here is called for. What we previously called a *state* of a Markov chain becomes a *stage* which itself contains a collection of *phases* that generate a non-exponential distribution. The token wanders from stage to stage governed by the *Italic* matrix, $\boldsymbol{P}$. As long as the token is in stage i, cells leave the system at rate λ_i, while the token visits the *phases* in stage i, spending a mean time of $1/(\mathbf{M_i})_{kk}$ in each phase, k, and then traveling to another phase with probability determined by **Bold-faced** matrix, $\mathbf{P_i}$. The overall process describing the departure of cells (which, remember, forms the *arrival process* to a router or packet switch which services cells at the rate ν) is denoted by $\mathcal{C}aligraphic$ matrices, $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{B}}$, $\boldsymbol{\mathcal{P}}$, etc.

Let m be the number of stages. The transition matrix for the *generalized* Markov chain is

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \mathbf{P_1} + \mathbf{q}_1' P_{11}\, \mathbf{p}_1 & \mathbf{q}_1' P_{12}\, \mathbf{p}_2 & \cdots & \mathbf{q}_1' P_{1m}\, \mathbf{p}_\mathrm{m} \\ \mathbf{q}_2' P_{21}\, \mathbf{p}_1 & \mathbf{P_2} + \mathbf{q}_2' P_{22}\, \mathbf{p}_2 & \cdots & \mathbf{q}_2' P_{2m}\, \mathbf{p}_\mathrm{m} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{q}_\mathrm{m}' P_{m1}\, \mathbf{p}_1 & \mathbf{q}_\mathrm{m}' P_{m2}\, \mathbf{p}_2 & \cdots & \mathbf{P_m} + \mathbf{q}_\mathrm{m}' P_{mm}\, \mathbf{p}_\mathrm{m} \end{bmatrix} \tag{5.10}$$

where a component $\boldsymbol{\mathcal{P}}_\mathrm{ij}$ is itself a matrix of dimension $m_i \times m_j$. All square matrices of this type, have dimension $K \times K$, where

$$K = \sum_{i=1}^{m} m_i$$

Next, define the auxiliary matrices,

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{M}_2 & \cdots & \mathbf{O} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{M}_\mathrm{m} \end{bmatrix}; \quad \boldsymbol{\mathcal{B}}_0 = \begin{bmatrix} \mathbf{B}_1 & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{B}_2 & \cdots & \mathbf{O} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{B}_\mathrm{m} \end{bmatrix}. \tag{5.11}$$

As a direct generalization of (5.8),

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} \lambda_1 \mathbf{I}_1 & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \lambda_2 \mathbf{I}_2 & \cdots & \mathbf{O} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{O} & \mathbf{O} & \cdots & \lambda_m \mathbf{I}_\mathrm{m} \end{bmatrix}. \tag{5.12}$$

The generator of the generalized Markov chain can be calculated from

$$\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{P}}),$$

and the generator of the interdeparture times of the cells is

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{Q}} + \boldsymbol{\mathcal{L}}.$$

Although it was our intention to generalize the MMPP, since $\boldsymbol{\mathcal{P}\varepsilon'} = \boldsymbol{\varepsilon'}$, this process satisfies the properties of an MMPP. So, considered as an m-dimensional process, we can think of this as a generalization. But as a $K-$dimensional process it is a *restricted* MMPP. So much for definitions.

By direct manipulation, it can be shown that $\boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{B}}_0 - \boldsymbol{\mathcal{B}}_0\langle\, \boldsymbol{P}\,\rangle$, and thus

$$\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{B}}_0 + \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{B}}_0\langle\, \boldsymbol{P}\,\rangle,$$

where for any $m \times m$ matrix $\boldsymbol{W}$, $\langle\, \boldsymbol{W}\,\rangle$ is an $K \times K$ matrix of the form

$$\langle\, \boldsymbol{W}\,\rangle := \begin{bmatrix} W_{11}\,\boldsymbol{\epsilon}'_1\,\mathbf{p}_1 & W_{12}\,\boldsymbol{\epsilon}'_1\,\mathbf{p}_2 & \cdots & W_{1m}\,\boldsymbol{\epsilon}'_1\,\mathbf{p}_\mathrm{m} \\ W_{21}\,\boldsymbol{\epsilon}'_2\,\mathbf{p}_1 & W_{22}\,\boldsymbol{\epsilon}'_2\,\mathbf{p}_2 & \cdots & W_{2m}\,\boldsymbol{\epsilon}'_2\,\mathbf{p}_\mathrm{m} \\ \cdots & \cdots & \cdots & \cdots \\ W_{m1}\,\boldsymbol{\epsilon}'_\mathrm{m}\,\mathbf{p}_1 & W_{m2}\,\boldsymbol{\epsilon}'_\mathrm{m}\,\mathbf{p}_2 & \cdots & W_{mm}\,\boldsymbol{\epsilon}'_\mathrm{m}\,\mathbf{p}_\mathrm{m} \end{bmatrix}; \quad \text{with} \quad \boldsymbol{\varepsilon'} = \begin{bmatrix} \boldsymbol{\epsilon}'_1 \\ \boldsymbol{\epsilon}'_2 \\ \cdots \\ \boldsymbol{\epsilon}'_m \end{bmatrix}. \tag{5.13}$$

$\boldsymbol{\epsilon}'_\mathbf{i}$ is a column vector of m_i 1's. Since $\boldsymbol{P\epsilon'} = \boldsymbol{\epsilon'}$, it follows that $\langle\, \boldsymbol{P}\,\rangle\boldsymbol{\varepsilon'} = \boldsymbol{\varepsilon'}$. This algebraic procedure of *embedding* $m \times m$ matrices into $K \times K$ matrices has special properties. For instance, let $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ be any two $m \times m$ matrices. Then

$$\langle\, \boldsymbol{W_1}\,\rangle\langle\, \boldsymbol{W_2}\,\rangle = \langle\, \boldsymbol{W_1 W_2}\,\rangle,$$

and in general

$$\langle\, \boldsymbol{W}\,\rangle^n = \langle\, \boldsymbol{W}^n\,\rangle.$$

This can be taken further. Let $\boldsymbol{\mathcal{X}}$ be any $K \times K$ matrix, written in *block* form:

$$\boldsymbol{\mathcal{X}} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \mathbf{X}_{13} & \cdots & \mathbf{X}_{1\mathrm{m}} \\ \mathbf{X}_{21} & \mathbf{X}_{22} & \mathbf{X}_{23} & \cdots & \mathbf{X}_{2\mathrm{m}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{X}_{\mathrm{m}1} & \mathbf{X}_{\mathrm{m}2} & \mathbf{X}_{\mathrm{m}3} & \cdots & \mathbf{X}_{\mathrm{mm}} \end{bmatrix}$$

where $\mathbf{X}_{\mathrm{ij}}$ is an $m_i \times m_j$ matrix. Then it follows by direct substitution that

$$\langle\, \boldsymbol{W_1}\,\rangle\boldsymbol{\mathcal{X}}\langle\, \boldsymbol{W_2}\,\rangle = \langle\, \boldsymbol{W_1 X W_2}\,\rangle$$

where $\boldsymbol{X}$ is an $m \times m$ matrix with *scalar* elements

$$(\boldsymbol{X})_{ij} = \mathbf{p_i}\,\mathbf{X_{ij}}\,\boldsymbol{\epsilon}'_\mathbf{j} = \sum_{k=1}^{m_i}\sum_{l=1}^{m_j}(\mathbf{p_i})_k(\mathbf{X_{ij}})_{kl}.$$

In other words, the product of three matrices of dimension K $[\langle\, \boldsymbol{W_1}\,\rangle\boldsymbol{\mathcal{X}}\langle\, \boldsymbol{W_2}\,\rangle]$, can be carried out by multiplying three matrices of dimension m $[\boldsymbol{W_1 X W_2}]$ and then embedding the product $[\langle\, \boldsymbol{W_1 X W_2}\,\rangle]$. Two other algebraic properties are worth mentioning. If

$$\langle\, \boldsymbol{W_1}\,\rangle\langle\, \boldsymbol{W_2}\,\rangle = \langle\, \boldsymbol{W_2}\,\rangle\langle\, \boldsymbol{W_1}\,\rangle, \quad \text{then} \quad \langle\, \boldsymbol{W_1 W_2}\,\rangle = \langle\, \boldsymbol{W_2 W_1}\,\rangle.$$

Also,

$$\langle\, \boldsymbol{W_1}\,\rangle + \langle\, \boldsymbol{W_2}\,\rangle = \langle\, \boldsymbol{W_1} + \boldsymbol{W_2}\,\rangle.$$

This allows for a computationally simpler expression of $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{Y}}$. It can be shown that

$$\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{B}}^{-1} = \left[\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{D}}_0\langle\, (\boldsymbol{I} - \boldsymbol{PD}_0)^{-1}\boldsymbol{P}\,\rangle\right]\boldsymbol{\mathcal{V}}_0\boldsymbol{\mathcal{D}}_0, \tag{5.14}$$

where $\boldsymbol{\mathcal{D}}_0$, $\boldsymbol{\mathcal{V}}_0 = \boldsymbol{\mathcal{B}}_0^{-1}$, and $\boldsymbol{\mathcal{L}}$ are all block diagonal, and commute with each other, $\boldsymbol{\mathcal{D}}_0$ being

$$\begin{aligned} \boldsymbol{\mathcal{D}}_0 &:= (\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{V}}_0)^{-1} \\ &= \begin{bmatrix} (\mathbf{I}_1 + \lambda_1 \mathbf{V}_1)^{-1} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & (\mathbf{I}_2 + \lambda_2 \mathbf{V}_2)^{-1} & \cdots & \mathbf{O} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{O} & \mathbf{O} & \cdots & (\mathbf{I}_\mathrm{m} + \lambda_m \mathbf{V}_\mathrm{m})^{-1} \end{bmatrix}. \end{aligned} \tag{5.15}$$

Also,

$$\boldsymbol{D}_0 = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & d_m \end{bmatrix} \tag{5.16}$$

where

$$d_i = \Psi_i[(\mathbf{I_i} + \lambda_i \, \mathbf{V_i})^{-1}] = \mathbf{p_i}[(\mathbf{I_i} + \lambda_i \, \mathbf{V_i})^{-1}] \, \boldsymbol{\epsilon_i}'.$$

But $\boldsymbol{\mathcal{V}}_0 \, \boldsymbol{\mathcal{D}}_0 \, \boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{D}}_0$, so (5.14) leads to

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{V}}\boldsymbol{\mathcal{L}} = \left[\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{D}}_0 \langle \, (\boldsymbol{I} - \boldsymbol{P}\boldsymbol{D}_0)^{-1} \boldsymbol{P} \, \rangle \right] (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{D}}_0).$$

It follows from (5.5) that d_i is $F^*(\lambda_i)$, the Laplace transform of the distribution function represented by $< \mathbf{p_i}, \mathbf{B_i} >$ evaluated at $s = \lambda_i$. It also has a physical meaning. It is the probability that the token will enter and leave stage i without any cell leaving the system.

The K vector $\boldsymbol{\wp}$ can be found directly from the m vector, $\boldsymbol{u} = [u_1, \, u_2, \, \cdots, \, u_m]$, satisfying $\boldsymbol{u}\boldsymbol{P} = \boldsymbol{u}$. First define

$$\langle \boldsymbol{u} \rangle := [u_1 \mathbf{p}_1, \, u_2 \mathbf{p}_2, \, ..., \, u_m \mathbf{p}_m] \, .$$

Then, using (5.9), it can be verified that

$$\boldsymbol{\wp} = c \, \langle \boldsymbol{u} \rangle \boldsymbol{\mathcal{L}} \boldsymbol{\mathcal{V}}_0, \quad \text{where} \quad c^{-1} = \sum_{i=1}^{m} \lambda_i \, u_i \, \mathrm{E}\,(X_i)\,,$$

satisfies $\boldsymbol{\wp} \, \boldsymbol{\mathcal{Y}} = \boldsymbol{\wp}$, with $\boldsymbol{\wp}\boldsymbol{\varepsilon}' = 1$. Also,

$$\langle \boldsymbol{u} \rangle \, \langle \boldsymbol{P} \rangle = \langle \boldsymbol{u} \rangle.$$

5.4.6 Burst Processes - ON-OFF Model

In this section we begin to discuss a concrete application. More general models are given in Part II. Consider a single data *source* which intermittently puts data onto a telecommunications line leading to a switch, or some such communications node. Suppose for definiteness that data-files are the objects being sent, and that they are sent one file at a time. The source prepares a file for sending, then sends it, and then prepares another file for sending. The preparation entails retrieving a file from local storage and breaking the

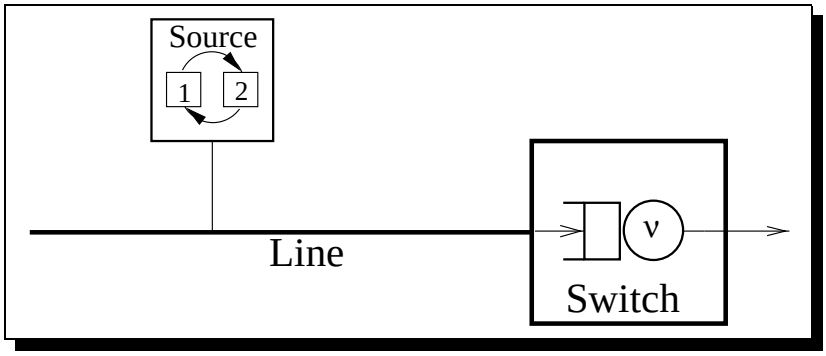


Figure 5.3: *Figure of source, line, and switch:* Token alternately visits two stages (each stage is a black box) in the source, which puts *cells* on the line at rate λ_p whenever the token is at the ON stage. The cells then queue up at the switch. In this paper it is assumed that the switch is exponential, with service rate ν.

file into *cells*. Let the mean time for this preparation be Z. The cells are transmitted at rate λ_p. Assuming no *background traffic*, the data rate while the source prepares the next file is $\lambda_b = 0$. This constitutes an *ON-OFF process*. The collection of cells is a *burst*. Let S_f be the random variable denoting the size of a file, s_c be the size of a cell, and n_p be the average number of cells in a burst, then

$$n_p = \mathrm{E}\left(\left\lceil \frac{S_f}{s_c} \right\rceil\right).$$

The mean time, $\overline{x_p}$, for a burst is

$$\overline{x_p} = \frac{n_p}{\lambda_p}.$$

If the distribution of file-size lengths is known, then a model can be set up. For then the ON-time distribution can be calculated. If $F_{S_f}(x)$ is the distribution of file sizes (in bytes, say), and $F_X(t)$ is the distribution of the time length of a burst (in seconds), then the two are related by

$$F_X(t) = F_{S_f}(s_c\,\lambda_p\,t).$$

There are 2 stages ($m = 2$), with $\lambda_1 = 0$ (OFF-time), and $\lambda_2 = \lambda_p$. The mean throughput (overall number of cells per second that arrive at the switch) is λ where

$$\lambda = \frac{n_p}{Z + \overline{x_p}} = \frac{n_p \lambda_p}{Z\lambda_p + n_p}.$$

Given the service rate, ν for the switch, its utilization parameter is

$$\rho = \frac{\lambda}{\nu}.$$

For the calculations presented here, we have assumed that the OFF-time is exponentially distributed ($m_1 = 1$) with a mean OFF-time of Z, but that assumption can easily be changed. We have made another assumption that is *not* so easy (but still possible) to modify, namely that the time between cells is exponentially distributed (at rate λ_p).

The question remains as to what ON-time distribution to pick. For instance, if one picks it to be exponential, then the process reduces to a renewal process, whatever the OFF-time

distribution is. For if $m_2 = 1$ then $\boldsymbol{\mathcal{L}}$ (and therefore, $\boldsymbol{\mathcal{Y}}$) is of rank one, and from (5.6) and the discussion following it, the cell interdeparture times are independent. Let $F_X(\cdot)$ be represented by $< \mathbf{p_2}, \mathbf{B_2} >$, with $m_2 = \mathrm{Dim}(\mathbf{B_2})$. Then we have a *one-burst*, ON-OFF analytic model with parameters λ_p, Z, ν, n_p (or $\overline{x_p}$), ρ (or λ), etc.

5.5 Distribution of ON-Times

As mentioned in the Introduction, the usual collection of test functions will not do to model telecommunication traffic. In this section we discuss this in more detail, and then present a class of functions, the TPT distributions, that have been used successfully by our group in analytically modeling these systems.

5.5.1 Experimental Measurements

[Leland et al., 1994] reported measurements of Ethernet traffic in a now-famous paper. They showed that the number of packets in a given time interval varied greatly from one interval to the next, no matter how big or little the intervals were, from 10 milliseconds to 100 seconds. The variability of the data looked the same over a range of 5 orders of magnitude (the full range of their data). They called it *self-similar* traffic. The classic MMPP models failed to simulate that behavior. In fact, the data resembled *sub-Gaussian noise*, normally generated by α*-stable* distributions. They later calculated the autocorrelation functions, $\varrho(k)$ [see (5.7) and surrounding material], and indeed, it was closer to $1/k^c$ than to ϕ^k. Those measurements have been reproduced many times by those, and other, researchers (although albeit, not all the time).

[Lipsky et al., 1992] and [Hatem, 1997] looked at UNIX file-size distributions at the University of Connecticut, and found that the largest files satisfied a power law for 5 orders of magnitude. That is,

$$R_X(x) = Pr(X > x) \longrightarrow \frac{c}{x^\alpha}$$

with $\alpha \approx 1.5$. Note that if this behavior extended to ∞ the distribution would have infinite variance! In general, $\mathrm{E}(X^\ell)$ is finite for $\ell < \alpha$, and is infinite for $\ell \geq \alpha$. They also showed that a renewal process with *power-tail* interarrival times can simulate self-similar data, but with a different appearance.

More recently [Crovella & Bestavros, 1997] in measuring *web traffic* at Boston University have seen this behavior in WWW document sizes, UNIX file sizes, files requested, transmitted files, and unique transfers. They also separated out audio, video, image and text files. Although these all have very large variances, only the text files had power-tail behavior.

The conclusion one must come to based on these measurements is that any model of telecommunications traffic that includes transfer of text and/or data files must in some way use distributions that generate self-similar behavior. On the other hand, a model focusing on video traffic only, should use some other distribution. Given that we are interested in file transfers, our obvious choice was to assume that ON-times have power-tail [or at least *truncated power-tail* (TPT)] distributions.

5.5.2 A Matrix Representation of a PT Distribution

Except for some numerical difficulties, it is not much more difficult to carry out discrete event simulations of PT (or TPT) distributions than for distributions with variance. But PT distributions are not well suited for analytic modeling, since the Laplace transform of such functions are not known, except by numerical integration. However, Greiner, Jobmann and Lipsky have introduced a family of ME distributions that approach PT with increase of dimension (or *Truncation parameter*, T). One can find a detailed description of these functions in [Greiner et al., 1999]. We only summarize their results here.

Let X_T, and $< \mathbf{p}_{\mathrm{T}}, \mathbf{B}_{\mathrm{T}} >$ as given below, be the random variable and $T-$dimensional representation of a *Truncated Power-tail* reliability function, $R_T(x)$.

$$0 < \theta < 1, \quad \gamma > 1$$

$$\mathbf{p}_{\mathrm{T}} = \frac{1-\theta}{1-\theta^T}[1,\, \theta,\, \theta^2,\, \cdots .\theta^{T-1}]$$

$$\mathbf{M}_{\mathrm{T}} = \mathbf{B}_{\mathrm{T}} = \mu \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1/\gamma & 0 & \cdots & 0 \\ 0 & 0 & 1/\gamma^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1/\gamma^{T-1} \end{bmatrix},$$

where μ is a positive constant that can be chosen to set the mean of the distribution according to (5.19). Then

$$\mathbf{V}_{\mathrm{T}} = \frac{1}{\mu} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \gamma & 0 & \cdots & 0 \\ 0 & 0 & \gamma^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & \gamma^{T-1} \end{bmatrix}.$$

Since $\mathbf{B}_{\mathrm{T}}$ is diagonal, it is easy to show from (5.3) that

$$R_T(x) = \frac{1-\theta}{1-\theta^T} \sum_{j=0}^{T-1} \theta^j \, e^{-\mu\, x/\gamma^j}. \tag{5.17}$$

Its pointwise limit exists, giving

$$R(x) := \lim_{T\to\infty} R_T(x) = (1-\theta) \sum_{j=0}^{\infty} \theta^j \, e^{-\mu\, x/\gamma^j}. \tag{5.18}$$

The expectation value follows directly from (5.4)

$$\mathrm{E}(X_T) = \mathbf{p}_{\mathrm{T}}\, \mathbf{V}_{\mathrm{T}}\, \boldsymbol{\epsilon}'_{\mathrm{T}} = \frac{1}{\mu} \frac{1-\theta}{1-\theta^T} \sum_{j=0}^{T-1} (\gamma\theta)^j = \frac{1}{\mu} \frac{1-\theta}{1-\theta^T} \frac{1-(\gamma\theta)^T}{1-\gamma\theta} \longrightarrow \frac{1}{\mu} \frac{1-\theta}{1-\gamma\theta}, \tag{5.19}$$

and the higher moments are given by

$$\mathrm{E}(X_T^\ell) = \ell!\, \mathbf{p}_{\mathrm{T}}\, \mathbf{V}_{\mathrm{T}}^\ell\, \boldsymbol{\epsilon}'_{\mathrm{T}} = \frac{\ell!}{\mu^\ell} \frac{1-\theta}{1-\theta^T} \sum_{j=0}^{T-1} (\gamma^\ell\theta)^j.$$

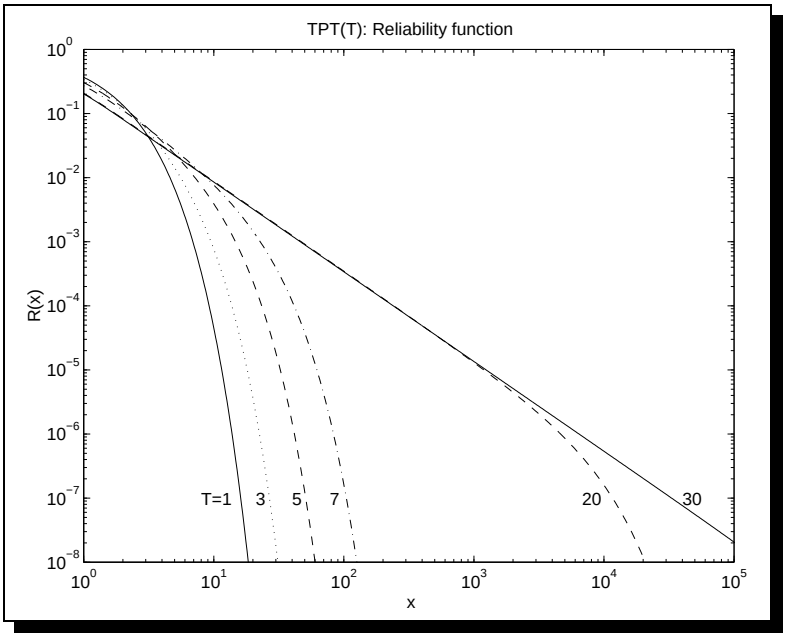


Figure 5.4: $R_T(x)$ *versus* x *on log-log scale:* $R_T(x)$ is taken from (5.17), with $T = 1$, $3, 5, 7, 20$, and 30. The linear portion grows ever longer with increasing T.

Define α to be the number that satisfies $\gamma^\alpha \theta = 1$ (or $\alpha := -\ln(\theta)/\ln(\gamma)$). As long as $\ell < \alpha$ (i.e., $\gamma^\ell \theta < 1$), the limit $T \longrightarrow \infty$, can be taken to give

$$\lim_{T \longrightarrow \infty} \mathrm{E}(X_T^\ell) = \frac{\ell!}{\mu^\ell} \frac{1-\theta}{1-\gamma^\ell \theta} \quad (\gamma^\ell \theta < 1).$$

But if $\ell \geq \alpha$ ($\gamma^\ell \theta \geq 1$) then the sum diverges and

$$\lim_{T \longrightarrow \infty} \mathrm{E}(X_T^\ell) = \infty \quad (\gamma^\ell \theta \geq 1).$$

That is, $R(x)$ from (5.18) satisfies the *moment property* for power-tail distributions. In Figure 5.4 $\log(R_T(\,\cdot\,))$ is plotted versus $\log(x)$ for various values of T. Clearly, the straight line characteristic for true PT distributions appears over an ever wider range with increasing T.

5.6 1-Burst Example with TPT ON-Times

Even without doing any calculations, certain properties can be ascertained for the 1-Burst/M/1 system. Consider a class of systems in which λ_p and Z are varied together so as to keep the load on the switch fixed (ρ, ν, and λ are held constant). As λ_p increases unboundedly (and Z approaches n_p/λ), the last cell in any given burst arrives at the queue before the first cell of the burst has been processed. Thus the system approaches the M/M/1 queue with *bulk arrivals*. The problem can also be translated into an M/G/1 queue where "G" is the distribution of the time it would take to process all the cells in a burst, and the computed response time is that for the time until the last cell in the burst is processed. The *buffer problem* (i.e., the probability that an arriving cell will see a full buffer) is somewhat more complicated to compare since the M/G/1 model assumes that

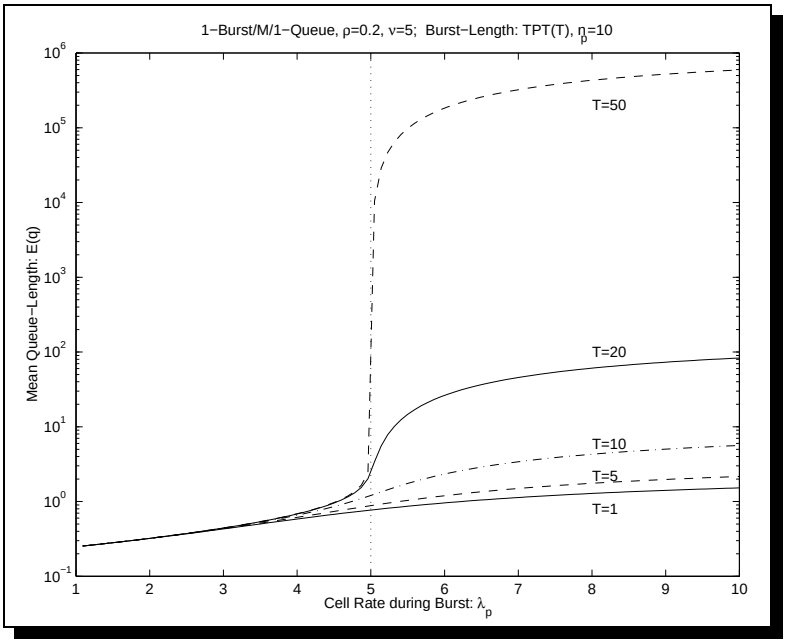


Figure 5.5: *Log of Mean Queue Length at Switch Versus Cell Burst Rate:* λ_p and Z (mean OFF-time) are varied together so that mean arrival rate to switch, λ, and its utilization, ρ, are fixed at 1.0 and 0.2 respectively.

all cells from a burst remain in the queue until the last one is processed. In any case, this is the worst case scenario.

At the other extreme, consider what happens as λ_p becomes smaller. It's minimum value occurs when $Z = 0$ (below that value, the average arrival rate, λ can no longer be maintained). Then there is no time elapsed between bursts. Therefore a continuous Poisson stream of rate λ occurs, reducing the system to a simple M/M/1 queue. This would be the best case scenario.

For low λ_p, performance of the switch is independent of burst-size distribution. But for large λ_p, it is highly dependent, certainly on the distribution's second moment. In the intermediate region, knowledge of the PDF is critical.

We have carried out mean queue length calculations for TPT(T) ON-times, where $T = 1$, $5, 10, 20$, and 50. The results are displayed in Figure 5.5. Note that it is the log of the queue length that is plotted versus λ_p. We have chosen a rather small $\rho = 0.2$ and small mean burstsize of $n_p = 10$. Every point on every curve is for the same ρ and n_p. This means that in all cases the switch is idle 80% of the time, and queue lengths as large as 20 would imply that more than 2 bursts are waiting to be served. As surmised in the preceding paragraphs, if the rate at which cells are sent to the switch is low, then queue lengths are negligible, and are insensitive to the ON-time distribution. But as the rate increases, the mean queue length literally *blows up* if $T > 10$ at $\lambda_p = 5$. This corresponds to an ON-time arrival rate that saturates the switch during a burst. Clearly, the system cannot recover during the OFF-times when T is large. Part II will discuss this in more detail, and show curves for overflow probabilities for the same family of TPT's as well as other distributions with the same mean and variance.

5.7 References

[Crovella & Bestavros, 1997] Crovella, M. and Bestavros, A. (1997). Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846.

[Fiorini, 1997] Fiorini, P. (1997). *Modeling Telecommunication Systems with Self-Similar Data Traffic.* Dissertation, University of Connecticut.

[Fiorini et al., 1995] Fiorini, P., Lipsky, L., Hsin, W.-J., and van de Liefvoort, A. (1995). Auto-Correlation of Lag-k For Customers Departing From Semi-Markov Processes. Technical Report TUM-I9506, Institut für Informatik, Technische Universität München.

[Greiner et al., 1999] Greiner, M., Jobmann, M., and Lipsky., L. (1999). The Importance of Power-tail Distributions for Modeling Queueing Systems. To appear in 'Operations Research: Telecommunications Area'.

[Hatem, 1997] Hatem, J. (1997). *Comparison of Buffer Usage Utilizing Single and Multiple Servers in Network Systems with Power-tail Distributions.* Dissertation, University of Connecticut.

[Leland et al., 1994] Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. (1994). On the self-similiar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15.

[Lipsky, 1992] Lipsky, L. (1992). *Queueing Theory: A Linear Algebraic Approach.* MacMillan and Company, New York.

[Lipsky et al., 1992] Lipsky, L., Garg, S., and Robbert, M. (1992). The Effect of Power-tail Distributions on the Behavior of Time Sharing Computer Systems. In *1992 ACM SIGAPP Symposium on Applied Computing*, Kansas City, MO. March 1992.

[Lipsky & Schwefel, 1999] Lipsky, L. and Schwefel, H.-P. (1999). N-Burst Processes as Analytic Models of Self-Similar Traffic in Telecommunications Systems – II. Specific System Models and Performance Results. In Greiner, M. and Jobmann, M., editors, *Stochastic Modeling of High-Speed Networks.* CS Press, München.

[Neuts, 1981] Neuts, M. F. (1981). *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* Johns Hopkins University Press, Baltimore, MD.

[Schwefel, 1997] Schwefel, H.-P. (1997). Modeling of Packet Arrivals Using Markov Modulated Poisson Processes with Power-Tail Bursts. Master's thesis, Institut für Informatik der Technischen Universität München.

6 N-Burst Processes as Models of Self-similar Traffic in Telecommunication Systems – II. Specific System Models and Analytic Results*

Lester Lipsky† **& Hans-Peter Schwefel**‡

Abstract

The pure ON/OFF-Process of the 1-Burst Model is extended to the N-Burst model that allows up to N sources to transmit simultaneously. The start- and end-points of these transmissions (bursts) can be described by embedded closed queueing systems of M/G/N//K-type. The flexibility of that concept allows to model a large variety of physical systems especially when introducing load-dependent servers in the embedded queueing system.

In the second part, the self-similar N-Burst variant is used as the arrival process to analyze the performance of network components that are modeled by N-Burst/M/1-Queues. The analytic results for mean Cell Delay and Cell Loss Probability reveal distinct critical utilization-values (blow-up points) at which the mean queue-length and the buffer-overflow probabilities radically increase. A second parametric study varies the cell-rate within individual bursts: If the intra-burst rate is very low, the model approaches an M/M/1 system. With very high intra-burst rate, the model approaches a bulk-arrival system. Network components are expected to operate in the intermediate region, where the blow-up points are contained. In that region, modeling on cell-level, as done in the N-Burst model, is shown to be essential. Furthermore, experiments with other high-variance distributions clearly indicate that the knowledge of only a few moments of the burst-length distribution is not sufficient for an analysis of the performance characteristics of the traffic model.

6.1 N-Burst Models

The 1-Burst process as introduced in [Lipsky & Schwefel, 1999] (referred to as *Part I* hereafter) is a pure ON/OFF Process as shown in Figure 6.1. It is a reasonable model for the physical scenario of a network line which is used by a single source. For multiple

* A major part of this research was funded by Deutsche Telekom AG.

† Department of Computer Science and Engineering, University of Connecticut, Storrs, U.S.A.
E-mail: lester@brc.uconn.edu

‡ Institut für Informatik, Technische Universität München, Germany
E-mail: schwefel@in.tum.de

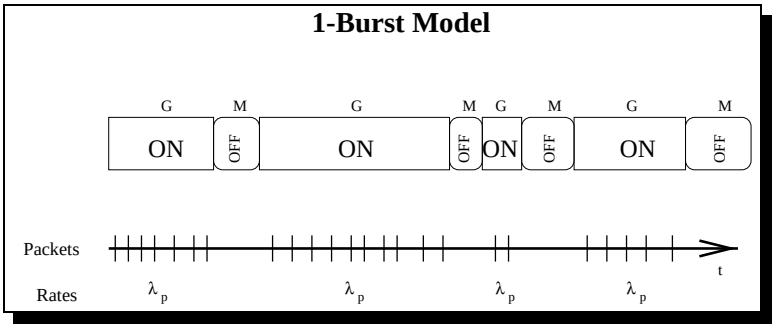


Figure 6.1: *ON/OFF Process of 1-Burst Model:* Generally distributed ON-times with cell-rate, λ_p, followed by exponentially distributed OFF-times.

sources, various scenarios are possible, depending on the network topology and the type of network: First, the network line can be exclusive, in which case any additional bursts have to wait to get on the line (1-Burst with waiting bursts). The resulting process is still of ON/OFF type, however, two or more ON-periods can occur without any OFF-periods in between.

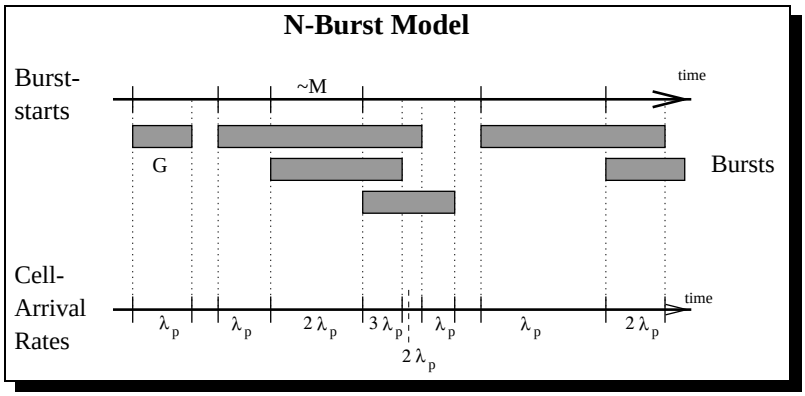


Figure 6.2: *Overlapping Bursts of N-Burst Model:* Up to N bursts can contribute their individual cell rate, λ_p, to the aggregated cell-stream. There are various possibilities for the process that describes the burst-starts. However, a large class of physical meaningful scenarios can be described uniformly. See text.

The second scenario is a network line that is shared by up to N bursts all of which contribute packets/cells[1] with rate λ_p, shown in Figure 6.2. In that case, the aggregated cell stream is a modulated Poisson Process with rates $(i \cdot \lambda_p)$, $i = 0, ..., N$. Hereafter, the pure ON-OFF model will be referred to as *1-Burst* while the N potentially overlapping bursts are captured by the *N-Burst* model. The accurate description of the start- and end-points of the individual bursts will be done uniformly for both models by the following:

The modulating processes in both scenarios can be described by closed queuing systems of M/G/N//K-type which is shown in Figure 6.3: The K customers in the closed system are the potential bursts. The number of active bursts on the line corresponds to the number of active servers in subsystem S_1. The distribution of the burst-length (holding-time) thereby is the general (matrix-exponential) distribution of the service times of the N servers at S_1.

[1] The traffic models are not specificly designed for a particular network architecture or transmission protocol. However, we use the ATM-terminology (i.e. *cells*) hereafter.

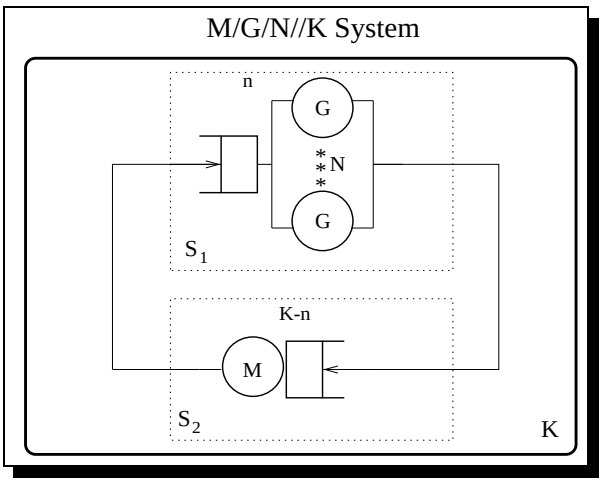


Figure 6.3: *Closed Queueing system of type M/G/N//K as Modulating Process of N-Burst Model:* N parallel identical servers at S_1. K potential bursts in the system. The Poisson Cell-Rate of the N-Burst model is modulated by the number of active servers at S_1.

Hereafter, the notation $[M/G/N//K]_{\lambda_p}$ will be used for the N-Burst process which is the modulated Poisson Process that later on describes the inter-arrival times of cells to some network component downstream. The system in square brackets is the modulator, the cell-rate during bursts, λ_p, can be omitted.

Subsystem S_2 of Figure 6.3 is the burst-arrival generating stage. It is defined to be a load-dependent exponential delay-stage. The flexibility of the load-dependence factors, $\gamma(k)$, where $k = K - n$ is the number of customers at subsystem S_2 given that there are n customers at S_1, extends the range of potential physical models. Two particular choices of $\gamma(\cdot)$ have significant physical meanings:

1. In case of a finite number of K sources, a so-called *infinite server stage* (also called *time-sharing stage*) with $\gamma(k) = k$ has to be used for the burst generating subsystem S_2: The higher the number n of active sources is, the lower the burst-start rate becomes, because less idle sources are left that can initiate new bursts. Using Kendall's notation, that special embedded system will be called $IS/\mathrm{G}/N//K$, where 'IS' stands for *Independent Sources*. The resulting rate of the delay-stage at S_2 is $\gamma(k)/Z = k/Z$ when k customers are at S_2, i.e. k of the K sources are not transmitting cells. Z is the mean think-time of the individual source.

   In the special case of no bursts ever waiting in the queue at S_1 ($N = K$), the resulting cell-stream is the superposition of K independent ON/OFF streams of the kind as produced by the 1-Burst [M/G/1//1] system (Fig. 6.1). The mean rate of that specific model follows:

$$\lambda^{[IS/G/N//N]} = \frac{N\, n_p}{Z + \overline{x}} = \frac{N}{Z\,\frac{\lambda_p}{n_p} + 1} \cdot \lambda_p \,, \tag{6.1}$$

   where n_p is the mean number of cells per burst, $\overline{x} = n_p/\lambda_p$ is the mean burst-length, and Z is the mean length of the OFF-periods of the individual sources, as already introduced in Part I.

2. For a WAN-like situation with a large pool of potential sources ($\gg K$; now, K is not the number of sources but the maximum number of bursts in the system), the rate of burst-starts is not affected by the number of active sources. In that scenario, no load-dependence applies ($\gamma(\cdot) \equiv 1$) and the burst-start process is a Poisson process with constant rate, μ_M. If more than K sources try to be active simultaneously, the additional bursts are lost. The mean cell rate in case of no waiting bursts ($N = K$) follows from the solution of the M/M/N//N queue (see e.g. [Kleinrock, 1975]):

$$\lambda^{[M/G/N//N]} = \frac{H(a, N-1)}{H(a, N)}\, a\, \lambda_p\,, \tag{6.2}$$

$$\text{with}^2 \quad a := \mu_M \overline{x}, \quad \text{and} \quad H(a, N) := \sum_{i=0}^{N} \frac{a^i}{i!}\,.$$

Note that (6.2) is based on the steady-state mean queue-length of the M/**M**/N//N queue. The mean queue-length of the M/**G**/N//N system in fact only depends on the mean of the service-time, not on its distribution (see [Lipsky, 1992]). Thus, the general server can be replaced by an exponential server for the purpose of calculating mean queue-lengths in the modulator.

The [M/G/1//1] and the [IS/G/1//1] system are identical for $\mu_M = 1/Z$, since the load dependence in S_2 only matters for $K > 1$.

Note that the N-Burst Process uses a closed queueing system to describe the burst-arrivals and their duration. There is no modeling of network components yet. If bursts are waiting for the line, which can happen for $K > N$, then they are waiting at the sources. Actual queueing of individual cells will be done in the next section by an SM/M/1-queue that uses the modulated Poisson arrivals of cells generated by the N-Burst [x/G/N//K] process as input.

Introducing Load-Dependent Holding-Times: Finally, the range of possible physical scenarios can be further extended by introducing load-dependence in the holding-time of the bursts at S_1. Thereby, back-off behavior which e.g. is a consequence of several congestion control mechanisms can be modeled: As soon as an additional burst starts, the individual rate of the currently active bursts reduces to $(\beta(n) \cdot \lambda_p)$, where n is the new number of active bursts. In the LAQT model, this reduction of the individual cell-rates requires a modification of the cell-rate matrix $\boldsymbol{\mathcal{L}}$ (see Part I for its definition). Furthermore, since the number of cells in the burst (which has mean n_p) must not change, the reduction of the cell-rate by the factor $\beta(n)$ must be compensated for by modifying the holding time by a factor $1/\beta(n)$. Thus, the servers at subsystem S_1 in Figure 6.3 have to be load dependent as well which in its most general form leads to closed queueing-systems of type $\mathrm{M}_{ld[\gamma(\cdot)]}/\mathrm{G}_{ld[\beta(\cdot)]}/N//K$ for the particular modulating processes of the N-Burst model.

The use of matrix-exponential (ME) distributions for the holding times even allows for an implementation of the notion of *traffic mixes*, which are a peculiarity of modern heterogeneous networks. Distinct traffic classes have different back-off behavior. When using a block-diagonal ME-representation of the burst-length distribution with each block

[2] a would be the mean number of simultaneously active bursts, if no limitation existed, i.e. for $N \to \infty$.

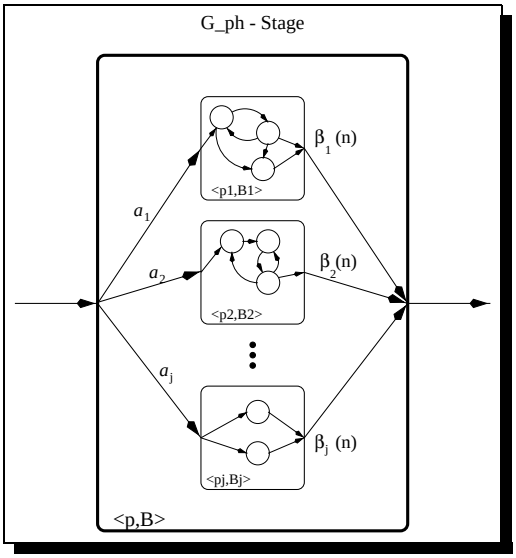


Figure 6.4: *Matrix-Exponential Representation of the Burst-Length Distribution for Mixes of Different Traffic Classes:* Each of the $m = 1, ..., j$ blocks represents a traffic class with proportion a_m in the overall traffic. Different load-dependence factors $\beta_m(n)$ can be assigned to model different back-off behavior.

$m = 1, ..., j$ corresponding to a traffic class, each traffic class can be assigned a distinct set of back-off factors, $\beta_m(n)$, as shown in Figure 6.4.

Though the aforementioned load-dependent generalizations of the N-Burst directly correspond to physical systems, the performance evaluation hereafter will focus on the [IS/G/N//K] model for reasons of simplicity. The choice of the burst-length distribution (the 'G' in the [x/G/N//K] system) turns out to be critical for the performance characteristic of network components. When using Power-Tail distributions or their truncated cousins (see Part I or [Greiner et al., 1999]), (truncated) long-range dependence in the inter-cell times can be observed. The parametric studies in the next section use this approach to gain insight into the impact of self-similar traffic on QoS parameters in network components.

6.2 Performance Results for Self-Similar N-Burst

Assuming that the N-Burst process is capable of describing real network traffic closely enough, the performance of network components such as ATM-switches can now be evaluated in exact analytic models by using the LAQT-techniques that were presented in Part I.

The network component is modeled as a queue with exponential server of rate ν and infinite buffer, i.e. an SM/M/1 model with the N-Burst process as specific *Semi-Markov* (SM) arrival process. The performance parameters of interest will be the mean Cell Delay (CD), the Cell Delay Variation (CDV), and the Cell Loss Probability (CLP) of a so-called backup-system[3]. The mean CD and the CDV are critical in particular for real-time applications such as video or voice traffic.

The evaluation of a finite SM/M/1/B_s loss-system with matrix-geometric methods is possible (see [Krieger et al., 1998]) but no results are shown here. However, the behavior

of the infinite-buffer system (investigated below) in principle also holds for the loss-system, with the restriction that the queue-length and consequently the Cell Delay are obviously bound by the buffer-size.

Modern ATM-Switches tend to have very large buffers (order of 10^4 to 10^6 cells). The advantage of Matrix-Geometric solutions is that their computational demand is independent of the buffer size, i.e. all computations are done in the state-space that is defined by arrival and service-process. For that reason, LAQT-methods seem to be most capable for analytic modeling of modern network components with large buffers.

When modeling ATM-switches, the service time of cells should really be deterministic. This could be modeled by employing Erlangian-k distributions for the service-time with the drawback of increasing the state-space by a factor k. However, for the interesting scenarios of potentially long queues[4], the exponential service time is expected to be a good enough approximation.

Section 6.2.1 discusses the computation of the aforementioned QoS parameters in detail. Thereafter, the results of two parametric studies are presented, while first observing the mean Cell Delay (Section 6.2.2) and then the Cell Loss Probability (Section 6.2.3). The two different parametric studies (vary truncation, T, in Section 6.2.2.1, and vary intra-burst cell-rate, λ_p, in Section 6.2.2.2) are conducted for three different types of burst-length distributions: Truncated Power-Tail Distributions (TPT(T)), Hyperexponential-2 distributions with same mean and variance, and Hyperexponential-2 distributions with same first three moments as the TPT(T) distribution. In addition, three different 1-Burst and 2-Burst processes (namely, [IS/G/1//1], [IS/G/1//2], and [IS/G/2//2]) are compared with each other in Section 6.2.2.1 and Section 6.2.3.1.

6.2.1 Matrix-Geometric Methods for SM/M/1 Models

Section 4.3 of Part I already mentioned the existence of a matrix geometric solution

$$\mathbf{r}(k) = \mathbf{r}(0)\boldsymbol{\mathcal{R}}^k, \quad \sum_{k=0}^{\infty} \mathbf{r}(k)\boldsymbol{\varepsilon}' = 1,$$

for the steady-state queue-length probabilities of an SM/M/1 queue.

The *Rate-Matrix*, $\boldsymbol{\mathcal{R}}$, has to satisfy the following quadratic matrix equation:

$$\nu\boldsymbol{\mathcal{R}}^2 - \boldsymbol{\mathcal{R}}(\boldsymbol{\mathcal{B}} + \nu\boldsymbol{\mathcal{I}}) + \boldsymbol{\mathcal{L}} = \mathbf{0}.$$

$\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{B}}$ define the SM arrival process as discussed in Section 4 of Part I. ν is the rate of the exponential server downstream from the semi-Markov generator/box and $\boldsymbol{\mathcal{I}}$ is the unit matrix. Efficient iterative algorithms for solving the matrix equation are discussed in [Latouche & Ramaswami, 1993], [Bini & Meini, 1996],

[3] The infinite buffer of the SM/M/1 model consists of a primary buffer of B_s cells and an infinite secondary buffer. The CLP is then the probability that an arriving cell has to go to the backup-buffer. As such it is an upper bound of the CLP in a finite SM/M/1/B_s loss-system.

[4] In classical models, long queues usually result from high utilization. This is not the case for our N-Burst as we will see later: Already for a low utilization (e.g. $\rho = 0.2$), arbitrary long mean queue-lengths can be observed.

and [Krieger et al., 1998]. However, the drawback of these iterative approaches for nearly self-similar arrival-processes is a widely varying demand of computation time due to the potentially huge resulting queue-lengths (see below and also [Lipsky & Schwefel, 1999]).

More important than the scalar steady-state queue-length[5] distribution, $r(k) = \mathbf{r}(k)\boldsymbol{\varepsilon}'$, is the probability, $a(n)$, that an arriving cell will find n other cells at the server. This is given in [Fiorini, 1997] as

$$a(n) = (\wp \boldsymbol{\mathcal{V}} \boldsymbol{\varepsilon}')\, \boldsymbol{\pi}\, [(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{R}})\boldsymbol{\mathcal{R}}^n\, \boldsymbol{\mathcal{L}}]\, \boldsymbol{\varepsilon}'.$$

For the definition of $\wp$, $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{B}}^{-1}$, and $\boldsymbol{\pi}$ see Section 4 of Part I.

By straightforward calculations, the mean queue-length at cell arrival times and its second moment follow:

$$\overline{q_a} = (\wp \boldsymbol{\mathcal{V}} \boldsymbol{\varepsilon}')\, \boldsymbol{\pi} \boldsymbol{\mathcal{R}} (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{R}})^{-1} \boldsymbol{\mathcal{L}} \boldsymbol{\varepsilon}'.$$

$$\overline{q_a^2} = (\wp \boldsymbol{\mathcal{V}} \boldsymbol{\varepsilon}')\, \boldsymbol{\pi} \boldsymbol{\mathcal{R}} (\boldsymbol{\mathcal{R}} + \boldsymbol{\mathcal{I}}) (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{R}})^{-2} \boldsymbol{\mathcal{L}} \boldsymbol{\varepsilon}'. \tag{6.3}$$

The Cell Delay (CD) is equal to the system time, S, of an individual cell. Since an arriving cell will see n cells already in the system with probability $a(n)$, the distribution of S is a mixture of Erlangian-$(n+1)$ distributions with probability $a(n)$, $n = 0, 1, 2, \ldots$. Consequently, the first two moments of the cell delay follow by straightforward calculations:

$$\mathrm{E}\,(S) = \frac{\overline{q_a} + 1}{\nu}, \quad \mathrm{E}\left(S^2\right) = \frac{\overline{q_a^2} + 3\overline{q_a} + 2}{\nu^2}\ . \tag{6.4}$$

The quadratic coefficient of variation of the CD-distribution,

$$C^2(S) := \frac{\mathrm{E}\,(S^2)}{\mathrm{E}\,(S)^2} - 1, \tag{6.5}$$

can be used as a measure of the Cell Delay Variation (CDV)[6]. The CDV is only mentioned here without being the focus of the studies in this paper.

As mentioned above, the Cell Loss Probability (CLP) will be approximated in the infinite SM/M/1 queueing model by the probability that the arriving cell will find B_s or more cells already in the queue, which has a compact solution when using the LAQT-methods:

$$Pr(n \geq B_s) = \sum_{n=B_s}^{\infty} a(n) = (\wp \boldsymbol{\mathcal{V}} \boldsymbol{\varepsilon}')\, \boldsymbol{\pi}\, \boldsymbol{\mathcal{R}}^{B_s} \boldsymbol{\mathcal{L}}\, \boldsymbol{\varepsilon}'. \tag{6.6}$$

6.2.2 Performance Results for Mean Cell Delay

Two parametric studies will be conducted to gain insight into the particular behavior of the first moment of the CD for the N-Burst model with truncated Power-Tail distributions with T phases (called TPT(T), see Part I) as burst-length distribution: The first study starts off with an N-Burst model with exponentially distributed burst-lengths ($T = 1$)

[5] Here, queue-length also includes the cell in service.

[6] The normalized 95%-quantile would be another possibility of describing the CDV.

and successively increases the number of phases, T, of the TPT distribution while keeping all other parameters of the model constant.

The second study varies the intra-burst cell-rate, λ_p, of particular N-Burst models. The mean cell-rate, λ, is kept constant by adjusting the mean OFF-time (think-time), Z, of the sources. So far, some results of this study for the 1-Burst [M/G/1//1] model were presented in Section 6 of Part I. Corresponding results for the 2-Burst model are discussed in Section 6.2.2.2 and 6.2.3.2.

All the calculations discussed herein use a TPT distribution that approximates a full PT with exponent $\alpha = 1.4$ in the reliability function. As a consequence, the variance of the PT with full tail will be infinite while the mean, $\overline{x}$, is finite. The mean number of cells in a burst is always fixed to $n_p = \overline{x}\,\lambda_p = 10$.

6.2.2.1 Impact of the PT Truncation

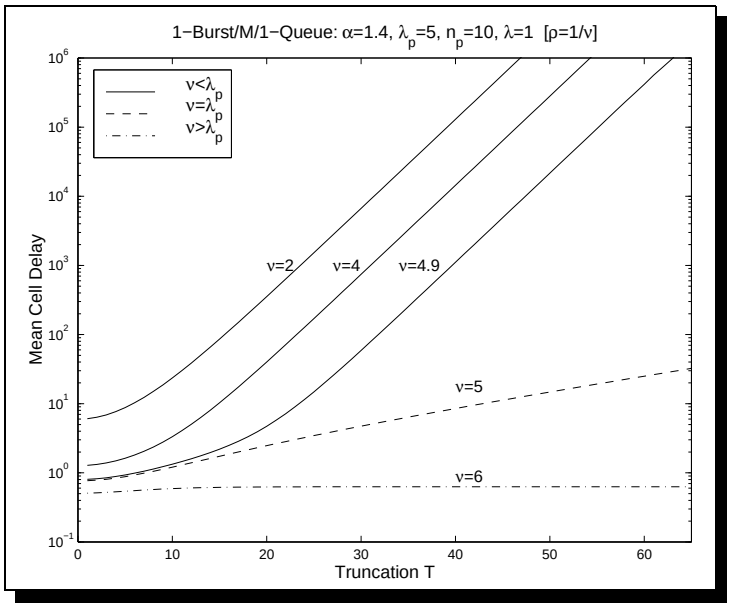


Figure 6.5: *Mean Cell Delay for 1-Burst [IS/G/1//1] System with Successively Increasing Truncation, T, of the Employed TPT distribution:* When the service rate is lower than the cell-rate during ON-time, $\nu < \lambda_p = 5$, which corresponds to $\rho > 0.2$ here, the mean CD grows geometrically with increasing T. The geometric factor (i.e. the slope of the curves) is in fact the same for all curves with $\nu < \lambda_p$. Note that the utilization is constant for each individual curve: $\rho = \lambda/\nu = 1/\nu$. Only the tail of the TPT distribution is expanded along the individual curves.

The TPT distributions enable us to evaluate the behavior of the SM/M/1 switch model while approaching self-similar traffic. Only when using an infinite number, T, of TPT-phases a full Power-Tail is obtained. In our finite physical world the introduction of a truncation for that full tail is reasonable due to existing physical limitations for e.g. file-sizes. The impact of this truncation is investigated hereafter.

Starting off with a single phase ($T = 1$) a series of TPT(T) distribution with increasing T but same mean $\overline{x} = n_p/\lambda_p$ is used as burst-length distribution for the N-Burst arrival process to an SM/M/1-queue. Figure 6.5 shows the exact analytic results for the mean

Cell Delay of a 1-Burst Model with no waiting bursts ([IS/G/1//1] system). The mean burst-length is kept constant at $\overline{x} = 2$, and the mean OFF-time Z is chosen such that the 1-Burst process has mean arrival rate $\lambda = 1$. For different service rates ν (which imply the utilization $\rho = 1/\nu$) curves of the mean Cell Delay for $\alpha = 1.4$ and $T = 1, ..., 65$ are shown. As long as the service-rate exceeds the cell-rate $\lambda_p = 5$ during the ON-times, the influence of the truncation is low, i.e. the results for exponentially distributed burst-lengths ($T = 1$) do not differ much from the results for PT burst-lengths ($T \to \infty$).

However, for $\nu \leq \lambda_p$ the situation changes radically. For large T, the mean Cell Delay grows geometrically with increasing T. All curves with $\nu < \lambda_p$ in Figure 6.5 show the same geometric factor (i.e. same slope in semi-logarithmic scale). Measurements of the slope of the curves in Figure 6.5 and in a large set of additional experiments of the same kind yield the following asymptotic relationship for the mean CD:

$$\mathrm{E}\,(S) \longrightarrow s_1 \left(\theta\gamma^2\right)^T, \quad \text{for} \quad \nu < \lambda_p \quad \text{and} \quad \theta\gamma^2 > 1. \tag{6.7}$$

Herein and hereafter, $s_i > 0$, $i = 1, 2, ...$ are independent of T (or $\overline{x_T}$, which will be introduced below). θ and γ are the parameters of the TPT distribution as introduced in Part I and [Greiner et al., 1999]. More meaningful is the exponent α of the Power-Tail:

$$\theta\gamma^2 > 1 \;\Leftrightarrow\; \alpha < 2.$$

Therefore, the geometric growth in Cell Delay only occurs for diverging variance of the burst-length distribution which happens for $\alpha < 2$. For $\alpha > 2$, the mean CD converges with $T \to \infty$.

What is the physical significance of this experiment? θ, T and consequently γ are auxiliary parameters for the ME-representation of TPT distributions. The last of the T phases of the TPT distribution has mean

$$\overline{x_T} = \frac{\gamma^{T-1}}{\mu_T}, \quad \text{where} \quad \gamma = \theta^{-1/\alpha}, \quad \mu_T = \frac{1-\theta}{1-\theta^T}\,\frac{1-(\theta\gamma)^T}{1-\theta\gamma}\,\frac{1}{\overline{x}}\ . \tag{6.8}$$

When $x > \overline{x_T}$, $R_T(x)$ drops off exponentially. That drop-off point, which is a characteristic of the burst-length distribution, provides a physical meaning to the truncation parameter T.

Upon plugging in the expressions for γ and μ_T in (6.8), it becomes clear that $\log(\overline{x_T}/\overline{x})$ and T have an asymptotically linear relationship for $\alpha > 1$ and large T:

$$\log\left(\frac{\overline{x_T}}{\overline{x}}\right) \;\longrightarrow\; a + b\,(T-1), \tag{6.9}$$

with the coefficients being

$$a = \log\left(\frac{1-\theta^{1-1/\alpha}}{1-\theta}\right), \quad \text{and} \quad b = \frac{1}{\alpha}\,\log\,\theta^{-1}\,.$$

As a consequence of (6.9), to appropriate scale, the curves of Figure 6.5 look (asymptotically) identical when plotting $\log_{10}$ of the mean CD versus $\log_{10}\,(\overline{x_T}/\overline{x})$ instead of plotting

against T itself. The geometric growth (6.7) of the mean CD then turns into a power-law growth for large $\overline{x_T}$:

$$\mathrm{E}(S) \longrightarrow s_2 \left(\frac{\overline{x_T}}{\overline{x}}\right)^{2-\alpha} \quad \text{for} \quad \nu < \lambda_p, \quad \text{and} \quad \alpha < 2. \tag{6.10}$$

Note that there are two conditions for an unbounded growth of the mean CD to happen[7]:

$$\alpha \leq 2 \quad \text{and} \quad \nu \leq \lambda_p =: \Lambda_1.$$

At the boundary values $\alpha = 2$ and $\nu = \lambda_p$ the mean CD still grows unboundedly, however the relation is different from (6.10).

Multiple Sources: 2-Burst

As mentioned in Section 6.1, the 1-Burst system [IS/G/1//1] corresponds to a single source, toggling between transmission and think-state.

When considering two potential bursts there are (at least) four possible systems: The 1-Burst [IS/G/1//2] model and the 2-Burst [IS/G/2//2] model correspond to two independent identical sources using a network line, which is exclusive in the former case and can handle both bursts simultaneously in the latter model. [M/G/1//2] and [M/G/2//2] correspond to a large pool of sources but no more than two bursts can be active or waiting for the line. Additional sources give up in that situation (see Section 6.1).

The 1-Burst models possess only two *levels* of operation: Either the process is in ON-state, generating cells with rate λ_p, or it is in OFF-state being idle. The unbounded Power-Law growth (blowup) of the mean Cell Delay occurs if cells during ON-level can saturate the server, i.e. $\nu \leq \lambda_p =: \Lambda_1$. Otherwise the CD and also other performance parameters converge very quickly and the TPT distributions even for high T do not give very different results from classical exponential burst-lengths.

Figure 6.6 compares the 1-Burst model with the 2-Burst model for the case of two independent sources (IS-system). The particular [IS/G/1//2] model of Figure 6.6 has a critical service rate of $\Lambda_1 = \lambda_p = 5$. Consequently, out of the set of curves for this model (plotted over the full range until $\overline{x_T} = 10^8$ in Figure 6.6) only the one for $\nu = 4$ shows the blowup of the mean CD.

Figure 6.6 also shows curves for an equivalent[8] 2-Burst [IS/G/2//2] model with the same service rates, ν, in the SM/M/1 model. Compared to the [IS/G/1//2] model, the mean CD is generally higher. But more important, the critical values for the service rate change and a Power-Law growth is observed for $\nu < 2\lambda_p = 10$. The asymptotic Power-Law behavior (6.10) with exponent (slopes) $2-\alpha = 0.6$ still holds for the curves with $\nu = 4$ and $\nu = 5.2$. However, a smaller exponent $3 - 2\alpha = 0.2$ is observed for $\nu = 5.7$ and $\nu = 8$.

In general, the N-Burst model has $N + 1$ levels of operation defined by i active bursts ($i = 0, ..., N$). When using (truncated) PT distributions extremely long bursts eventually

[7] Long-range dependence which is defined by a non-sumable auto-correlation function (see [Tsybakov & Georganas, 1997]) is only obtained for $\alpha \leq 2$ as well.

[8] Equivalent in the sense that the model parameters (n_p, λ_p) are the same except for the think-time Z which is adjusted in order to keep the overall mean cell rate $\lambda = 1$.

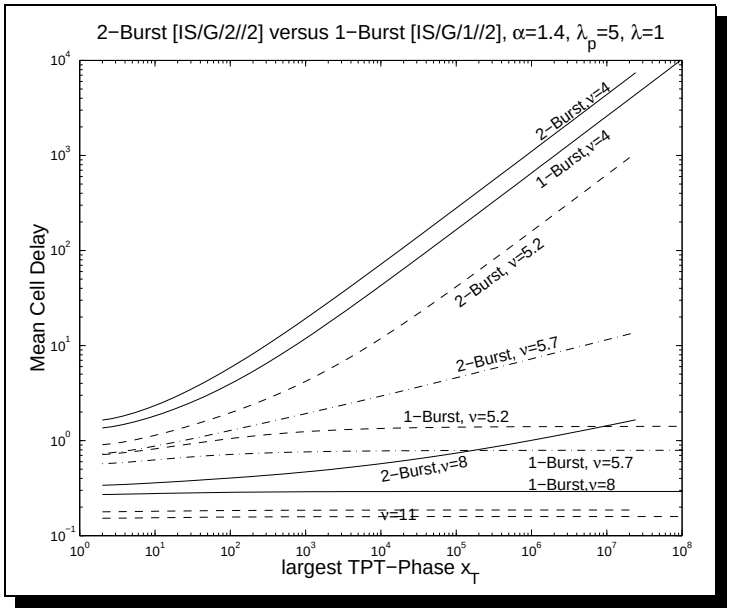


Figure 6.6: *Mean Cell Delay Comparing 1-Burst [IS/G/1//2] and 2-Burst [IS/G/2//2] Model for Successively Increasing Truncation:* Both models correspond to two independent sources. The curves that are shown over the full range until $\overline{x_T} = 10^8$ belong to the 1-Burst [IS/G/1//2] where the network line is exclusive. The choice of the specific model influences the values of the critical service rate Λ_i. Since $\Lambda_1 = 5$ is the only blowup point for the [IS/G/1//2] model while the [IS/G/2//2] model has $\Lambda_1 = 5.5$ and $\Lambda_2 = 10$, the curves for the service-rates $5 \leq \nu \leq 10$ show different behavior in the two models: They grow unboundedly for the [IS/G/2//2] model but converge for the [IS/G/1//2] model.

occur. During a long enough time-span, when i of the N bursts are permanently active, the mean cell-rate temporarily increases to:

$$\Lambda_i = i \cdot \lambda_p + \lambda^{[(N-i)-\mathrm{Burst}]}, \quad i = 0, 1, ..., N. \tag{6.11}$$

The first term of (6.11) is caused by the i bursts in a long ON-period while the second term is the mean cell-rate of the remaining potential bursts.

Using a server with rate $\nu \leq \Lambda_i$, Equation (6.11) implies, that i long-term active bursts together with the remaining $(N - i)$ bursts in average behavior[9] over-saturate the server temporarily. This over-saturation can be devastating if high-variance burst-length distributions are used, in particular PT-distributions.

For the [IS/G/2//2] model of Figure 6.6 the critical service rates are $\Lambda_1 = 5.5$ and $\Lambda_2 = 2\lambda_p = 10$. The asymptotic behavior of the mean CD for increasing truncation depends on the relation between the actual service-rate ν and the values of Λ_i. The Λ_i have the following order:

$$\lambda\,,\ \lambda_p\ \ \leq \Lambda_1 \leq \Lambda_2 \leq ... \leq \Lambda_N = N \cdot \lambda_p,$$

with all the inequalities being strict when $N > 1$.

[9] $\lambda^{[(N-i)-\mathrm{Burst}]}$ is the mean rate of an $(N-i)$-Burst process, that describes the behavior of the additional $(N - i)$ sources, e.g. for the [IS/G/N//N] model it is an [IS/G/$N - i$//$N - i$] model with same parameters n_p, λ_p, and Z.

For the [IS/G/N//N] model, which will be focused on herein, it follows from (6.11) and (6.1):

$$\Lambda_i = \left(i + \frac{N-i}{Z\frac{\lambda_p}{n_p}+1} \right) \lambda_p \;=\; i\lambda_p + \left(1 - \frac{i}{N}\right) \lambda\,. \tag{6.12}$$

Now, let i_0 be the smallest i, such that $\nu \leq \Lambda_i$,

$$i_0 := \min\left\{ i \,|\, \nu \leq \Lambda_i \right\}, \tag{6.13}$$

i.e. i_0 permanently active bursts would saturate the server but $(i_0 - 1)$ of them do not. Clearly, if $\nu > \Lambda_N$ then no saturation of the server can happen for that service rate. In that case, all *Quality of Service* (QoS) parameters are bounded and only little dependence on the actual burst-length distribution is observed (see also Section 6.2.2.2). Therefore, the case of $\nu > \Lambda_N$ is excluded in (6.13) and also excluded from the following discussion.

For the more interesting scenario of $\nu < \Lambda_N$, observations from Figure 6.6 and additional experiments show the following asymptotic behavior of the mean CD:

$$\mathrm{E}\,(S) \longrightarrow s_3 \left(\frac{\overline{x_T}}{\overline{x}} \right)^{1+i_0-i_0\alpha}, \quad \text{for} \quad \alpha < 1 + \frac{1}{i_0}. \tag{6.14}$$

However, the slope of the set of curves with $\Lambda_1 = 5.5 < \nu < \Lambda_N = 10$ in Figure 6.6 only approaches its asymptotic value $3 - 2\alpha = 0.2$ for rather high T.

As a consequence of (6.14), the critical values for α, where the Power-Law growth of the mean CD shows up, are $(1+1/i_0) = [2.0, 1.5, 1.33, ...]$. Therefore, the impact of the specific value of α depends strongly on i_0, which describes the relation between the service rate, ν, and the *blowup*-points, Λ_i. Thereby, the latter are mainly determined by the individual cell-rate within bursts, λ_p.

The asymptotic behavior of the Cell-Delay turned out to be independent of the specific type of model: The [IS/G/1//2] model and the [IS/G/2//2] model of Figure 6.6 result in almost parallel curves for the mean CD as long as the service-rate, ν, yields the same i_0 for both models. However, since the values of Λ_i depend on the specific modulator, the mean CD for $\nu = 5.2$, $\nu = 5.7$, and $\nu = 8$ behaves differently in these two models. Using an [M/G/2//2] model instead does not affect the general behavior of the mean CD either. But again, it changes the critical ν-values from $\Lambda_1^{(IS)} = 5.5$ for the [IS/G/2//2] system to $\Lambda_1^{(M)} \approx 5.85$ for the [M/G/2//2] system. Consequently, the only curve out of the ones shown in Figure 6.6 that strongly differs for the [M/G/2//2] model would be the one with $\nu = 5.7$, because $\Lambda_1^{(IS)} < 5.7 < \Lambda_1^{(M)}$.

Hyperexponential-2 Distributions for Burst-Lengths

To gain deeper insight into the particular impact of PT-distributed burst-lengths, another experiment replaces the TPT(T) distributions by other potentially high variance distributions. The simplest matrix-exponential distribution for that purpose is a Hyperexponential-2 (HYP-2) distribution that has the reliability function:

$$R_H(x) = p_1\, e^{-\mu_1 x} + (1-p_1)\, e^{-\mu_2 x}, \quad 0 < p_1 < 1, \quad 0 < \mu_1 < \mu_2\,.$$

The three free parameters p_1, μ_1, and μ_2 are used to match the first three moments of the HYP-2 distribution with the corresponding moments of the TPT(T) distribution. If only a two moment-match (i.e. mean and variance) is required, the probability of getting a very long burst, p_1, is chosen to a constant value beforehand. The smaller p_1 is chosen, the larger the third moment of the burst-length distribution turns out to be.

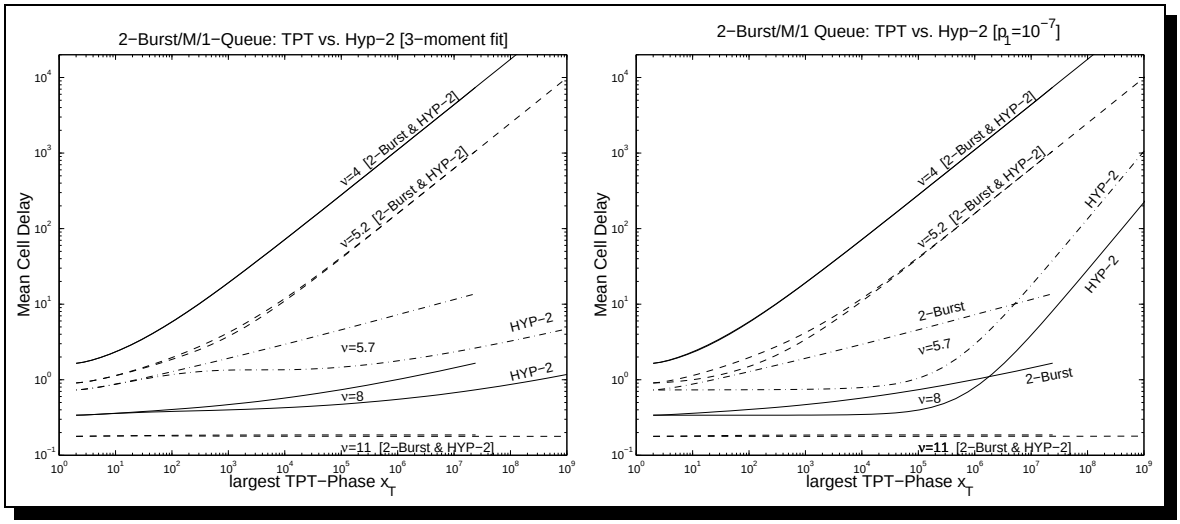


Figure 6.7: *Mean Cell Delay for 2-Burst [IS/G/2//2] System Comparing TPT with HYP-2 Distributed Burst-Lengths for Successively Increasing Truncation:* Left graph: Curves of mean CD for TPT distribution (plotted over shorter range until $\overline{x_T} \approx 10^7$) compared to 2-Burst model with 3-moment fit of HYP-2 distributions. Right graph: 3-moment fit is replaced by a 2-moment fit with constant $p_1 = 10^{-7}$ of the HYP-2 distribution.

Figure 6.7 compares in its left graph the three-moment match of a HYP-2 distribution with the TPT distribution in the [IS/G/2//2]/M/1 queue. The right graph shows a set of curves for a two-moment match with constant $p_1 = 10^{-7}$ of the HYP-2 burst-length distribution.

In both cases, the 2-Burst model with HYP-2 distributions turns out to be a close approximation in case of $\nu < \Lambda_1 = 5.5$ and $\nu > \Lambda_2 = 10$. However, in the intermediate region $\Lambda_1 < \nu < \Lambda_2$, a three moment match of the burst-length distribution is not sufficient. Only the asymptotic slope (exponent in (6.14)), $3 - 2\alpha = 0.2$, is identical for both sets of curves. If the HYP-2 distribution only matches two moments of the TPT(T) distribution, even that slope of the curves is affected (Figure 6.7, right).

6.2.2.2 Vary Burstiness λ_p

The experiments that were carried out for the 1-Burst process in Part I are now extended to two independent sources (2-Burst [IS/G/2//2] system): The cell-rate during bursts, λ_p, is varied while keeping the overall traffic constant by adjusting the mean OFF-time Z of both sources. Thus, λ_p and Z are varied simultaneously in order to hold $\lambda = 1$ constant while keeping $n_p = 10$, $\nu = 5$, and thus $\rho = 0.2$ constant. The necessary think-time Z of

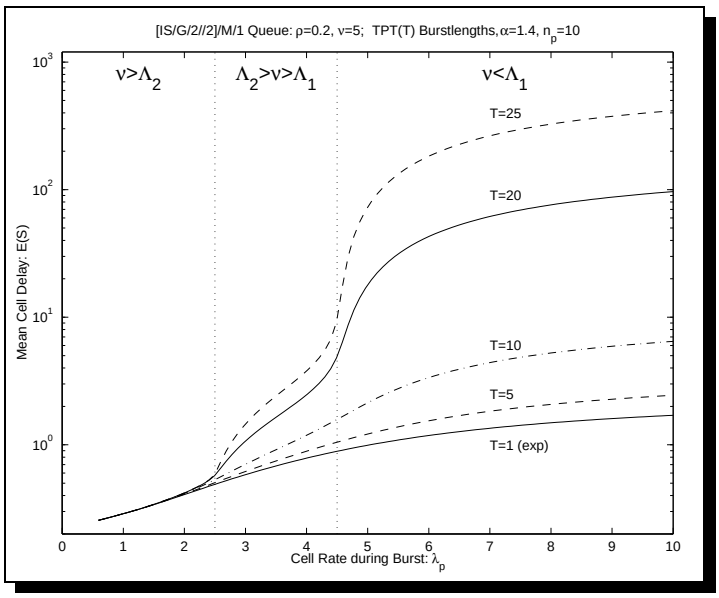


Figure 6.8: *Mean Cell Delay for 2-Burst [IS/G/2//2] System with Varying Intra-Burst Cell Rate:* The mean arrival-rate, $\lambda = 1$, and thus the utilization, $\rho = 0.2$, is thereby hold constant. Two blow-up points can be observed. The higher the truncation T gets, the more pronounced the blow-ups are.

the individual sources in the [IS/G/N//N] model follows from (6.1):

$$Z = n_p \cdot \left(\frac{N}{\lambda} - \frac{1}{\lambda_p} \right) .$$

For $Z = 0$, i.e. $\lambda_p = \lambda/N$, all sources generate smooth Poisson traffic in which case the SM/M/1 model reduces to the M/M/1 model, which does not show any blowups of the mean CD for $\rho < 1$.

When increasing λ_p as it was done in Figure 6.8, the critical Λ_i also increase (see (6.12)). Since the service rate $\nu = 5$ is kept constant, eventually

$$\Lambda_i \geq \nu \quad \text{for} \quad \lambda_p \geq \frac{\nu - \lambda}{i} + \frac{\lambda}{N},$$

in the [IS/G/N//N] model, and a blowup of the mean CD can be observed. The values of λ_p for which $\Lambda_i = \nu$ are marked by vertical dotted lines in Figure 6.8: The first blowup occurs at $\Lambda_N = \nu$, i.e. $\lambda_p = \nu/N = 2.5$, where both sources when active simultaneously saturate the switch. The second one happens at $\lambda_p = \nu - \lambda^{[IS/G/1//1]} = 4.5$ where one long-term active source plus average behavior of the second source saturates the switch.

The higher the truncation parameter, T, of the TPT distribution, the more pronounced are the blowups of the mean CD.

Hyperexponential-2 Distributions for Burst-Lengths

Again, it provides further insight into the significance of the TPT distributions to compare their impact in the N-Burst model with different distributions that have high variance, namely the HYP-2 distributions with matching first two or three moments. The most

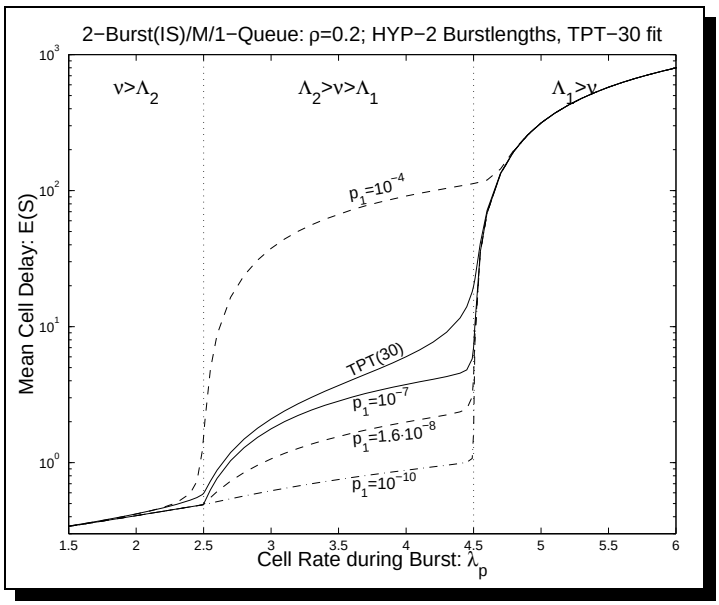


Figure 6.9: *Mean Cell Delay for 2-Burst [IS/G/2//2] System Comparing TPT and HYP-2 Distributed Burst-Lengths for Varying Intra-Burst Cell Rate:* All curves are for models with same first two moments of the burst-length distribution. For $p_1 = 1.6 \cdot 10^{-8}$ the third moment of the burst-length distribution also matches the third moment of the TPT(30) distribution.

pronounced blowup of Figure 6.8 is chosen for that experiment, i.e. the [IS/G/2//2] model with TPT(30) distributed burst-lengths.

The results in Figure 6.9 show that the intermediate region, $\Lambda_1 < \nu < \Lambda_2$, is highly sensitive to more than the first two moments of the specific burst-length distribution. All curves have same first two moments of the burst-length distributions, yet they are very different in that intermediate region. Even the 3-moment match ($p_1 = 1.6 \cdot 10^{-8}$) is not close to the TPT(30) distribution. Interestingly, a higher p_1 than the 3-moment fit can give a better approximation for the mean CD of the TPT model in Figure 6.9.

As soon as the last blowup point (where $\nu = \Lambda_1$ for $\lambda_p = 4.5$) is reached, the mean CD of all models with same variance of the burst-length distribution very quickly yields almost identical curves. So in that region, the first two moments of the burst-length distribution sufficiently determine the mean CD of the 2-Burst model. This is a stronger statement than the convergence for $\lambda_p \to \infty$. In that limit, the N-Burst/M/1 system reduces to a bulk-arrival system, whose mean CD can be shown to only depend on the first two moments of the bulk-size distribution. However, Figure 6.9 implies that the higher moments of the burst-length distribution already lose their influence shortly after crossing the last blowup point. For the plotted range of λ_p in Figure 6.9, the bulk-arrival model obviously does not apply yet because otherwise the curves would have to be nearly horizontal.

At the other end when λ_p is so small that $\nu > \Lambda_N$, the mean CD is not only insensitive to higher moments than the variance of the burst-length distribution (seen in Figure 6.9) but Figure 6.8 shows that the variance itself does not matter. In that region of operation, the used burst-length distribution is not significant and can in good approximation be replaced by a classical exponential distribution. Finally, at the very end for $\lambda_p = \lambda/N$, all burstiness is lost and the M/M/1 model[10] applies.

When replacing the [IS/G/2//2] model in the experiment by other 2-Burst models (e.g. [M/G/2//2]) only the location of the blowup points is different, yet the qualitative behavior of the mean CD remains unchanged.

6.2.3 Cell Loss Probability

So far, we observed the mean CD in two parametric studies with different (high-variance) burst-length distributions. All the experiments of Section 6.2.2 are now repeated while observing the CLP (approximated by the buffer-overflow probability in the backup model). The constant parameters ($\alpha = 1.4$, $n_p = 10$, $\lambda = 1$) are therefore complemented by a chosen buffer-size of $B_s = 10^4$ cells.

6.2.3.1 Impact of the PT Truncation

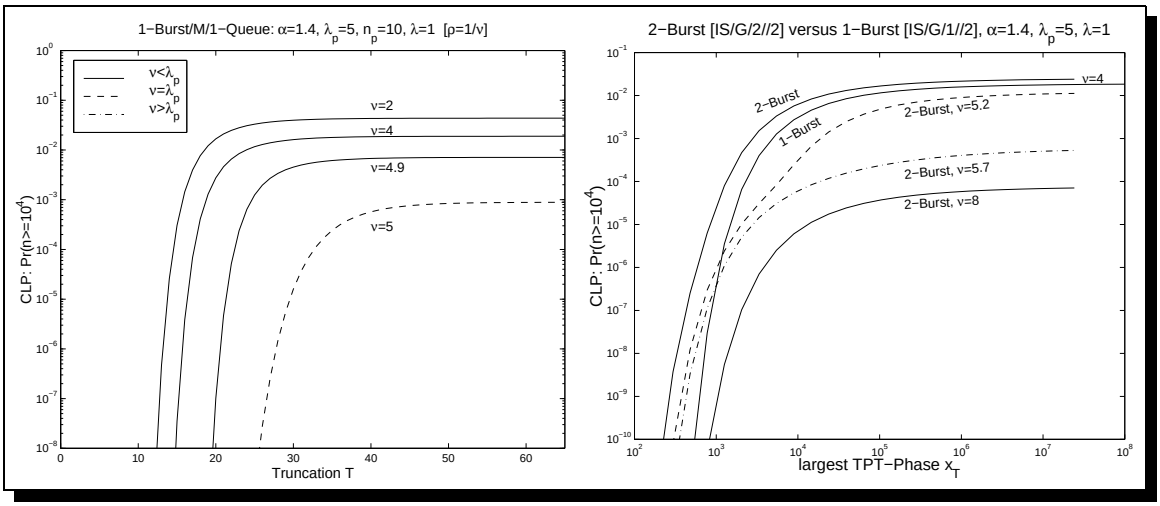


Figure 6.10: *CLP for Different N-Burst Systems with Successively Increasing Truncation, T:* Left: 1-Burst [IS/G/1//1] system with $\Lambda_1 = \lambda_p = 5$ as discussed in Figure 6.5. Right: [IS/G/1//2] model with $\Lambda_1 = 5$ in comparison to [IS/G/2//2] model where $\Lambda_1 = 5.5$ and $\Lambda_2 = 10$, as in Figure 6.6.

In contrast to the mean CD of Figure 6.5 there is no unbounded growth of the CLP for increasing truncation, T, in Figure 6.10: All curves converge to a value smaller than 1. The significance of the blowup value Λ_N in all the models of Figure 6.10 is that the CLP practically vanishes for $\nu > \Lambda_N$: Even if all N sources are permanently active, the cell rate $N\lambda_p$ does not exceed the service rate ν. Consequently, the CLP is bounded from above by the CLP $(\rho_M)^{B_s}$ in an M/M/1 model[11] with $\rho_M = N\lambda_p/\nu < 1$.

Though the curves for $\nu = 5.2$ and $\nu = 5.7$ of the [IS/G/2//2] model in Figure 6.10 are somewhat different, the remaining blowup-points other than Λ_N are not as clearly visible in that particular experiment when observing the CLP as they were for the mean CD.

[10] The M/M/1-queue has mean system time $E(S) = 1/(\nu - \lambda) = 0.25$.

[11] For the 1-Burst process in the left-hand side of Figure 6.10 and $\nu = 6$ the CLP is smaller than $(\rho_M)^{10^4} \approx 10^{-792}$.

Hyperexponential-2 Distributions for Burst-Lengths

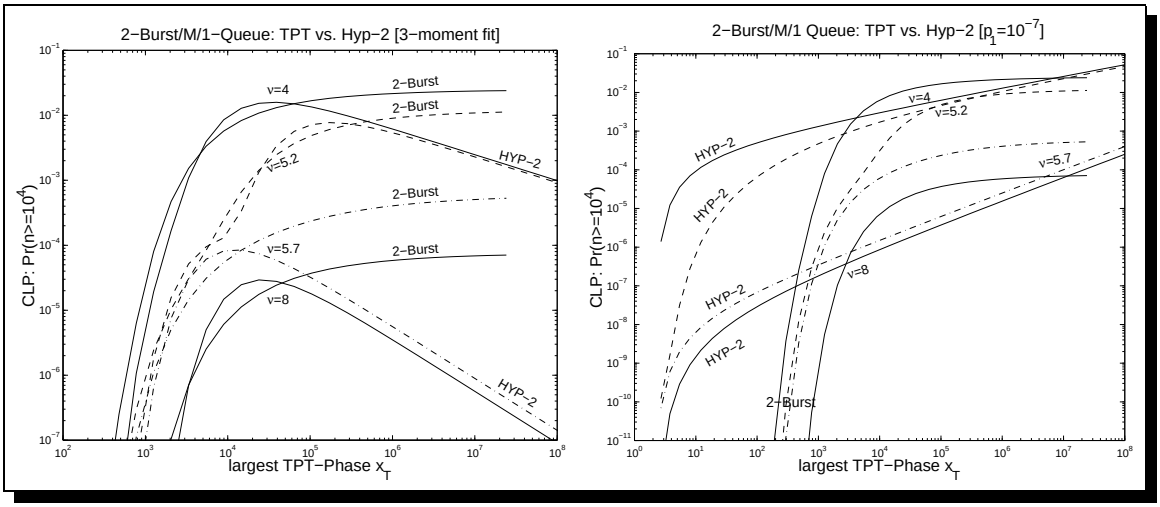


Figure 6.11: *CLP for [IS/G/2//2] System Comparing TPT with HYP-2 Distributed Burst-Lengths for Successively Increasing Truncation:* Left Graph: HYP-2 distribution matches first three moments of TPT(T) distribution. Then, the CLP shows a maximum with a Power-Law decay afterwards. Right Graph: HYP-2 distribution with $p_1 = 10^{-7}$ and matching first two moments.

Following the procedure of Section 6.2.2, the TPT(T) distributions are replaced by HYP-2 distributions with matching first two or three moments. Figure 6.11 compares three variations of the [IS/G/2//2] system, using TPT(T) distributions, three moment fits of a HYP-2 distribution, and two-moment fits with $p_1 = 10^{-7}$, respectively. Again, curves for $\nu > \Lambda_2 = 10$ do not appear in the graphs since the CLP in that situation is too low. Note that the observation in Figure 6.7 (i.e. the *mean CD* in the HYP-2 model represents a good approximation for the TPT model in case of $\nu < \Lambda_1 = 5.5$), does not hold any more for the CLP:

The 2-moment match of the HYP-2 distribution for constant $p_1 = 10^{-7}$ (Figure 6.11, right hand side) does not approximate the TPT model anywhere: It starts off with far too high CLP and grows with an asymptotic Power-Law with the slope depending on the blowup-region i_0.

The 3-moment fit (Figure 6.11, left) is more useful at first. However, a maximal CLP can be observed for some $\overline{x_T}$ in the set of curves for the HYP-2 distribution. After that maximum, the CLP drops off, again asymptotically guided by a Power-Law. Up to the point when the drop-off behavior starts, the CLP of the HYP-2 model is close to the CLP of the TPT model.

6.2.3.2 Vary Burstiness λ_p

Finally, the last set of experiments observes the CLP while varying the intra-burst cell rate, λ_p. The detailed description of this experiment is given in Section 6.2.2.2. As already mentioned there, at the left end at $\lambda_p = \lambda/N$, both sources have think-time $Z = 0$, thus they do not show any ON/OFF behavior. The M/M/1 model that applies in that case

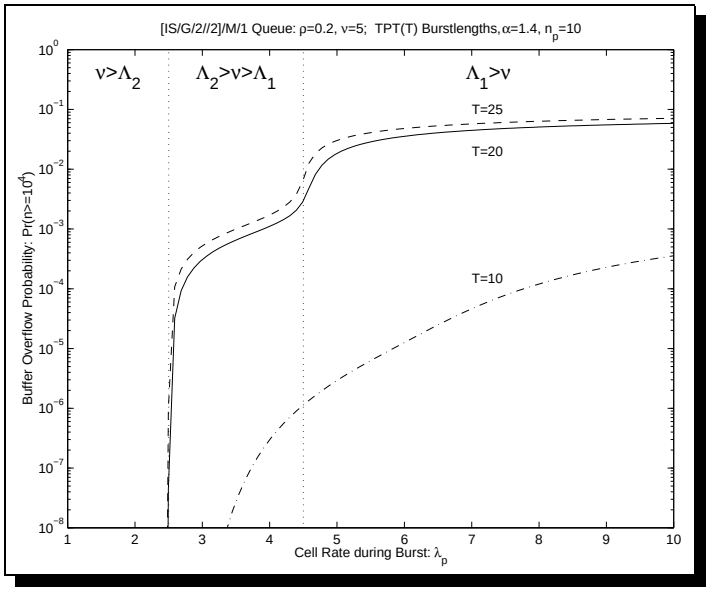


Figure 6.12: *CLP for [IS/G/2//2] System with Varying Intra-Burst Cell Rate:* Curves for $T = 1$ (maximal overflow probability $\approx 10^{-44}$) and for $T = 5$ (maximum $5 \cdot 10^{-16}$) not visible on that scale. Note that even though the utilization, $\rho = 0.2$, is rather low, the CLP can become almost 10% due to the degree of burstiness of the incoming cells.

has a CLP of $(\lambda/\nu)^{B_s} \approx 10^{-3000}$. Therefore, even on the semi-logarithmic scale of Figure 6.12 that value is by far below the plotted range.

When increasing λ_p, the sources show a higher degree of burstiness and the values of the critical service rates, Λ_i, increase. At $\lambda_p \geq 2.5$, $\nu \leq \Lambda_2$ and a distinct blowup can be seen for $T \geq 20$ in Figure 6.12. In fact, until $\lambda_p = 2.5$ the CLP is bounded by an M/M/1-queue with utilization $\rho_M = N\lambda_p/\nu < 1$ whose CLP is practically zero[12] for large buffers. Therefore the leftmost region, $\nu > \Lambda_2$, is not visible in the graph of Figure 6.12 and also not interesting in terms of performance analysis because QoS can be guaranteed in any case. The second blowup point at $\lambda_p = 4.5$ where $\nu = \Lambda_1$ is also clearly visible in Figure 6.12 for $T \geq 20$.

Hyperexponential-2 Distributions for Burst-Lengths

The blow-up points even show up more clearly when using HYP-2 distributed burst-lengths, shown in Figure 6.13. The CLP is almost horizontal in between the blowup points.

In contrast to the mean CD (Figure 6.9), the curves for the CLP do not converge to a single curve for $\nu < \Lambda_1$ ($\lambda_p > 4.5$). Knowing the first 2 (or 3 moments) of the burst-length distribution is thus not enough for estimating the CLP, even if $\nu < \Lambda_1$. The bulk-arrival model still applies for $\lambda_p \to \infty$. However, its CLP does not only depend on a few moments of the bulk-size distribution.

Also, the three moment fit yields a too low CLP compared with the TPT distribution. The

[12] For $\lambda_p = 2.4$ and $\nu = 5$, when both sources are permanently active, the resulting M/M/1 queue has a utilization $\rho_M = 0.96$, but for a large buffer of $B_s = 10^4$ cells its CLP is only $(\rho_M)^{B_s} \approx 10^{-177}$, i.e. practically zero.

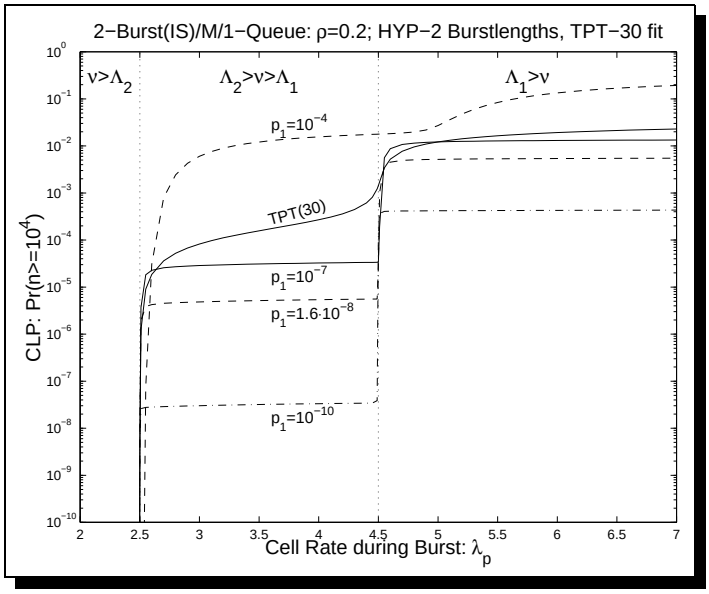


Figure 6.13: *CLP for 2-Burst [IS/G/2//2] System Comparing TPT with HYP-2 Distributed Burst-Lengths for Varying Intra-Burst Cell Rate:* The HYP-2 distribution with $p_1 = 1.6 \cdot 10^{-8}$ has the same first three moments as the TPT(30) distribution, but it still does not yield a good approximation of the CLP. The other curves use constant p_1 and match the first two moments of the TPT(30) burst-length distribution.

truncation of the TPT is $T = 30$, i.e. the last phase has mean $\overline{x_T}/\overline{x} = 6.2 \cdot 10^5$. According to the discussion around Figure 6.11, a better approximation could be expected for lower T with earlier drop-off.

6.3 Conclusion

The N-Burst model describes network traffic on cell-level as a superposition of up to N simultaneous bursts. It belongs to the more general class of modulated Poisson Processes. Different kinds of N-Burst Processes are introduced due to different physical scenarios yet they can be described uniformly by utilizing load-dependent closed queueing systems of M/G/N//K-type for the the modulating process. The analytic results in this paper concentrate on the N-Burst model with N independent, identical ON/OFF-sources ([IS/G/N//N] model) but the results are rather robust to changes of the model type. If the burst-length distribution in these models is a Power-Tail distribution ($R(x) \to c/x^\alpha$) with exponent $1 < \alpha < 2$, self-similar or long-range dependent traffic results.

The self-similar N-Burst model becomes amenable to LAQT-methods when employing the truncated TPT(T) distributions that show Power-Tail behavior in their reliability functions $R_T(x)$ up to $x \approx \overline{x_T}$ ($\overline{x_T}$ is the *range* of $R_T(x)$). The performance of network components such as ATM-Switches can then be evaluated analytically by utilizing SM/M/1 queues with the N-Burst process as the SM arrival process. Herein, the two QoS parameters Mean Cell Delay and Cell Loss Probability were observed under the influence of self-similar traffic.

The conducted parametric studies reveal the existence of distinct blowup-points, Λ_i (see (6.11)), where the resulting load temporarily equals the service rate for i long-term active

bursts. The behavior of the SM/M/1 model with a self-similar SM arrival process changes drastically at these blowup points. The relationship of the actual service-rate ν to the blow-up points Λ_i defines the model behavior: For $\nu < \Lambda_N = N\lambda_p$, a Power-Law growth (6.14) of the mean Cell-Delay for increasing range $\overline{x_T}$ of the truncated Power-Tail distribution can be observed under specific conditions on α.

Except for the extreme cases $\nu < \Lambda_1$ (a single burst can oversaturate the switch) and $\nu > \Lambda_N$ (no over-saturation even if all bursts are active[13], no QoS problems occur), which are not very likely for physical systems, the modeling on cell-level with the specific burst-length distribution is essential: Approximations such as bulk-arrival models or N-Burst models with other high-variance distributions for the burst-lengths are not eligible since they were shown to yield very different performance results.

6.4 References

[Bini & Meini, 1996] Bini, D. and Meini, B. (1996). On the solution of a nonlinear Matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:909–926.

[Fiorini, 1997] Fiorini, P. (1997). *Modeling Telecommunication Systems with Self-Similar Data Traffic.* Dissertation, University of Connecticut.

[Greiner et al., 1999] Greiner, M., Jobmann, M., and Lipsky., L. (1999). The Importance of Power-tail Distributions for Modeling Queueing Systems. To appear in 'Operations Research: Telecommunications Area'.

[Kleinrock, 1975] Kleinrock, L. (1975). *Queueing Systems. Volume I: Theory.* John Wiley & Sons, Inc.

[Krieger et al., 1998] Krieger, U., Naoumov, V., and Wagner, D. (1998). Analysis of a Finite FIFO Buffer in an advanced Packet-Switched Network. *IEICE Trans. Commun., VOL. E00-B*, 5.

[Latouche & Ramaswami, 1993] Latouche, G. and Ramaswami, V. (1993). A Logarithmic Reduction Algorithm for Quasi-Birth-Death Processes. *Journal of Applied Probability*, 30:650–674.

[Lipsky, 1992] Lipsky, L. (1992). *Queueing Theory: A Linear Algebraic Approach.* MacMillan and Company, New York.

[Lipsky & Schwefel, 1999] Lipsky, L. and Schwefel, H.-P. (1999). N-Burst Processes as Analytic Models of Self-Similar Traffic in Telecommunications Systems – I. Introduction and LAQT Background. In Greiner, M. and Jobmann, M., editors, *Stochastic Modeling of High-Speed Networks.* CS Press, München.

[Tsybakov & Georganas, 1997] Tsybakov, B. and Georganas, N. D. (1997). On Self-Similar Traffic in ATM Queues: Definitions, Overflow Probability Bound and Cell Delay Distribution. *IEEE/ACM Transactions on Networking*, 5(3):307–309.

[13] In ATM-switches, the condition $\nu > \Lambda_N = N\lambda_p$ only holds, when peak-rate reservation is used for connection setup. However, no multiplex-gain is achieved then.

Quadratically convergent algorithms for solving matrix polynomial equations

Dario A. Bini, Guy Latouche and Beatrice Meini

September 21, 2000

Abstract

The matrix equation $\sum_{i=0}^{n} A_i X^i = 0$, where the A_i's are $m \times m$ matrices, is encountered in many applications, in particular in the numerical solution of Markov chains which model queueing problems. We provide here a unifying framework in terms of Möbius' mapping to relate different resolution algorithms. This allows us to compare algorithms like Logarithmic Reduction and Cyclic Reduction, which extend Graeffe's iteration to matrix polynomials, and Matrix Sign Function iterations, which extend Cardinal's algorithm. We devise new iterative techniques having quadratic convergence and present numerical experiments.

1 Introduction

Let A_i, $i = 0, \ldots, n$ be $m \times m$ matrices and consider the matrix equation

$$\sum_{i=0}^{n} A_i X^i = 0. \tag{1}$$

It is encountered in many applicative fields, in particular in the analysis of queueing problems modeled by Markov chains of M/G/1 type [25, 24, 20, 30] in which case $I + A_1 \geq 0$, $A_i \geq 0$ for $i \neq 1$, and $\sum_{i=0}^{n} A_i + I$ is an irreducible stochastic matrix. In certain cases, $n = \infty$ and the left-hand side of (1) is a matrix power series; when $n = 2$, we have the so called Quasi-Birth-and-Death (QBD) problems which are of particular interest.

In Markov chains applications, one is interested in the minimal nonnegative solution of (1), and different techniques have been introduced in the literature to calculate that minimal solution, usually denoted by G. Algorithms based on functional iterations, having linear convergence, are analyzed

in [17, 29, 22, 11, 10]. A few methods with quadratic convergence have also been proposed: the Logarithmic Reduction (LR) technique is defined in [19] for the special case $n = 2$ and a similar iteration, the Cyclic Reduction (CR) method, is extended in [3, 4] to arbitrary values for $n \leq \infty$. The Matrix Sign Function (MSF) algorithm is introduced in [1] for general finite values of n. Finally, we also mention the doubling technique proposed in [21] and Newton's scheme analyzed in [18] — these, however, are less efficient and will not be considered further here. A recent survey on methods to determine the solutions of (1) in a general context is presented in [15].

We pursue here the analysis and critical comparison between CR and MSF performed in [23]. We show that Möbius' mapping $z(w) = (1 + w)/(1 - w)$ and its inverse $w(z) = (z-1)/(z+1)$ are the fundamental keys for expressing the relation between the CR and LR algorithms on the one hand, and the MSF algorithm on the other hand. More precisely, we show that the first two methods are particular extensions of Graeffe's iteration [26, 16] to matrix polynomials and that their quadratic convergence is due to the *implicit* use of the square function $S(z) = z^2$. The MSF algorithm, for its part, coincides with Cardinal's algorithm [9, 8], applied to a suitable Frobenius matrix and its convergence is due to the *explicit* use of Joukowski's function $J(x) = \frac{1}{2}(x + x^{-1})$. This is significant because the two functions are directly related by means of Möbius' mapping since $w(J(z)) = S(w(z))$, as one easily verifies.

The paper is organized as follows. We recall in Section 2 the main properties of Möbius' mapping, we analyze the correspondence between Joukowski's and the square functions and we explain why these are important in the context of the computation of the matrix G. In Section 3 we recall various iterative algorithms for factoring scalar polynomials (the iterations of Cardinal, Chebyshev, Graeffe and Sebastiao e Sylva) and relate them by means of Möbius' mapping.

In Section 4 we show how the LR, CR and MSF iterative procedures are based on generalizations to matrix polynomials of Graeffe's and Cardinal's procedures analyzed in Section 3, and we develop in Section 5 a new algorithm based on Chebyshev's iterations. We show that this new procedure is better than the MSF procedure. We give some numerical illustration in Section 6 and argue in Section 7 that the Cyclic Reduction procedure (and its LR variant) is the method of choice in the context of Markov chains, both in terms of its speed of convergence and of its numerical stability.

2 Möbius' Map

Denote by $\mathbb{C}^+$ and $\mathbb{C}^-$ the half-planes made up by the complex numbers with strictly positive and strictly negative real part respectively, and denote by $\mathcal{D}$ the open unit disk.

Möbius' map is characterized by the function

$$z(x) = (1+x)/(1-x), \tag{2}$$

of a complex variable, defined for $x \neq 1$. It maps the unit circle without the point 1 into the imaginary axis and the imaginary axis into the unit circle without the point -1. Moreover, it maps $\mathcal{D}$ into $\mathbb{C}^+$ and the complement of the closed unit disk into $\mathbb{C}^-$. Also, it maps $\mathbb{C}^-$ into $\mathcal{D}$ and $\mathbb{C}^+$ into the complement of the closed unit disk.

The inverse map

$$w(t) = (t-1)/(t+1) \tag{3}$$

defined for $t \neq -1$ has a very similar behavior: it maps the unit circle, without the point -1, into the imaginary axis and the imaginary axis into the unit circle without the point 1; the open unit disk $\mathcal{D}$ is mapped into $\mathbb{C}^-$, while the complement of the closed unit disk is mapped into $\mathbb{C}^+$; $\mathbb{C}^+$ is mapped into $\mathcal{D}$ and $\mathbb{C}^-$ into the complement of the closed unit disk.

These are important features of the transformation, and we shall write that the mapping $z(x)$ transforms the *coordinates of the unit circle* into the *coordinates of the imaginary axis*.

Remark 1 The mappings z and w are so similar that we might choose the mapping $w(x)$ for changing coordinates from the unit circle to the imaginary axis as it is done in the paper [1]. However, we prefer to use $z(x)$ since this leads to simpler expressions while leaving unchanged the substance of the matter. We return to this comment and motivate our choice in Remarks 2, 3 and 4 of later sections.

Mappings of roots of polynomials The two mappings are very useful when we have to recast algorithms by interchanging the roles of the unit circle and of the imaginary axis, as in the analysis of the stability of polynomials and of the inertia of matrices. Given a polynomial $p(x)$, we may either apply this transformation to the polynomial itself or to its variable x. In the latter case we obtain in a natural way the operator

$$\mathcal{P} : \Pi_n \to \Pi_n$$

defined on the set $\Pi_n = \{p(x) = \sum_{i=0}^{n} p_i x^i,\ p_i \in \mathbb{C}\}$ of polynomials of degree at most n having complex coefficients, by the following equation

$$\mathcal{P}(p(x)) \equiv q(t) = (1+t)^n p(w(t)). \tag{4}$$

If $p(x)$ has degree n and $p(x) = \prod_{i=1}^{n}(x - \xi_i)$, then

$$q(t) = \gamma \prod_{\substack{1 \le i \le n, \\ \xi_i \neq 1}} (t - \eta_i)$$

where $\eta_i = z(\xi_i)$, $\gamma = (-2)^k \prod_{\xi_i \neq 1}(1 - \xi_i)$, and k is the number of the zeros equal to 1. If, on the other hand, $p(x)$ has degree $h < n$ and $p(x) = \prod_{i=1}^{h}(x - \xi_i)$, then

$$q(t) = \gamma(t+1)^{n-h} \prod_{\substack{1 \le i \le h \\ \xi_i \neq 1}} (t - \eta_i).$$

Thus, if $p(x)$ has k roots in $\mathcal{D}$ and $n - k$ roots in the complement of the closed unit disk, then $q(t)$ has k roots in $\mathbb{C}^+$ and $n - k$ roots in $\mathbb{C}^-$.

Even though in principle there is no difference in the nature of the polynomials $p(x)$ and $q(t)$ in Π_n, we refer to Π_n as being in the *domain of the unit circle* when we want to point out its role as the domain of definition of $\mathcal{P}$, and in the *domain of the imaginary axis* when we refer to the co-domain where $\mathcal{P}$ takes its values. One motivation of this notational choice is that the operator $\mathcal{P}(\cdot)$ allows us to transform algorithms for polynomial and matrix computations defined in the domain of the unit circle into algorithms defined in the domain of the imaginary axis.

Mapping of operators Möbius' mapping may also be applied to operators. When one considers successive iterations of the square function

$$S(x) = x^2,$$

the complex numbers are naturally partitioned in three subsets: the unit circle, its interior $\mathcal{D}$ and its exterior. Sequences obtained from the recursion $x_{i+1} = S(x_i)$ quadratically converge to zero if $|x_0| < 1$ and to ∞ if $|x_0| > 1$; there is in general no convergence for starting points with $|x_0| = 1$.

For Joukowski's function

$$J(t) = \frac{1}{2}(t + t^{-1}),$$

the complex numbers are partitioned into $\mathbb{C}^+$, $\mathbb{C}^-$ and the imaginary axis: sequences obtained from the recursions $t_{i+1} = J(t_i)$ converge to 1 if $t_0 \in \mathbb{C}^+$, to -1 if $t_0 \in \mathbb{C}^-$ and fail to converge if t_0 is on the imaginary axis.

We say that Joukowski's function has its natural environment in the *domain of the imaginary axis* while that of the square function is in the *domain of the unit circle*. Observe that the fixed points 0 and ∞ of $S(\cdot)$ are mapped by $z(x)$ onto 1 and -1 respectively, and that the unit circle and the imaginary axis, where we have no convergence of the sequences obtained by means of the square function and Joukowski's function respectively, are mapped onto one another.

One proves by direct inspection the following properties:

$$w(-t) = \frac{1}{w(t)}, \qquad z(-x) = \frac{1}{z(x)}, \tag{5}$$

$$w(t^{-1}) = -w(t), \qquad z(x^{-1}) = -z(x), \tag{6}$$

$$w(J(t)) = S(w(t)), \qquad z(S(x)) = J(z(x)), \tag{7}$$

$$w(-S(t)) = J(w(t)), \qquad z(J(x)) = -S(z(x)). \tag{8}$$

Now, let us consider a function $f(x)$ of the variable x in the coordinates of the unit circle. We may apply the change of coordinates $t = z(x)$ both to the independent variable x and to f, thereby transforming $f(x)$ into the function $g(t) = z(f(w(t)))$ defined for t in the set of coordinates of the imaginary axis.

In this way, any function defined in the domain of the unit circle has its counterpart in the domain of the imaginary axis. In particular, we see from (5) that the function $f(x) = -x$ is associated to $g(t) = t^{-1}$ and that the function $f(x) = x^{-1}$ is associated to $g(t) = -t$. We see from (8) that the function $S(x)$ is associated to $J(t)$ and $J(x)$ is associated to $-S(t)$.

This is the key observation that allows us to recast algorithms defined in the domain of the unit circle into algorithms defined in the domain of the imaginary axis.

Remark 2 If we interchange the roles of $z(x)$ and $w(t)$, and use, like in [1], $w(x)$ as the function that maps the coordinates of the unit circle onto the coordinates of the imaginary axis, then the transformation $f(x) \to g(t)$ such that $g(t) = w(f(z(t))$, maps $S(x)$ into $1/J(t)$ and $J(x)$ into $S(t)$.

Spectrum of G The minimal nonnegative solution G of (1) has its natural environment in the domain of the unit circle because of the following property [12, 13]. Define $p(x) = \sum_{i=0}^{n} A_i x^i$ and denote by $\xi_1, \xi_2, \cdots$ the roots of $\det(p(x))$, ordered in increasing value of their modulus. The eigenvalues of G are the m smallest roots $\xi_1, \ldots, \xi_m$, the eigenvalue ξ_m with largest modulus is real, simple, and $\xi_m \leq 1$, furthermore, $|\xi_{m+1}|$ is strictly greater than ξ_m, and is at least equal to 1. Thus, it comes as no surprise that methods which

use repeated application of the square function will be useful in directly determining G.

The approach followed in [1] consists in applying Möbius' map (2), thus replacing (1) with the equation

$$\sum_{i=0}^{n} A_i (I - W)^{n-i} (I + W)^i = \sum_{i=0}^{n} U_i W^i = 0. \tag{9}$$

The solution W having eigenvalues in $\mathbb{C}^-$ provides the solution G by means of the formula $G = (I - W)^{-1}(I + W)$. The natural environment of W is in the domain of the imaginary axis, and it is clear why one should use Joukowski's function as the basis for iterative algorithms to compute W, as is done in [1].

3 Polynomial Factorization

The functions $S(x)$ and $J(t)$ are used to solve polynomial computational problems like approximating polynomial factors or polynomial roots.

Sebastiao e Sylva and Cardinal The algorithms of Sebastiao e Sylva [32] and of Cardinal [9] are two such applications. The first one uses the square function, the second uses Joukowski's function.

Given a polynomial $p(x)$, Sebastiao e Sylva's procedure is based on the sequence $\phi_{i+1}(x) = \phi_i^2 \bmod p(x)$, starting with an initial polynomial $\phi_0(x)$ (say, $\phi_0(x) = x$). Approximations to the factors of $p(x)$ containing equimodular zeros are determined by applying the Euclidean scheme to $p(x)$ and $\phi_i(x)$ for a large enough i. In Cardinal's algorithm, one generates the sequence of polynomials $\psi_{i+1}(x) = \frac{1}{2}(\psi_i(x) + \psi_i(x)^{-1}) \bmod p(x)$, starting from an initial polynomial $\psi_0(x)$ and one determines two factors of $p(x)$ containing the zeros with positive (respectively negative) real parts from $\gcd(\psi_i(x)+1, \psi_i(x)-1)$, for i large enough.

In matrix form, Sebastiao e Sylva's algorithm corresponds to applying the power method to the Frobenius matrix

$$F = \begin{pmatrix} 0 & 1 & & \bigcirc \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -p_0 & -p_1 & \dots & -p_{n-1} \end{pmatrix} \tag{10}$$

associated with the polynomial $p(x)$, and Cardinal's method corresponds to applying the matrix sign iteration to the same matrix. One easily verifies

from (8) that Cardinal's method may be interpreted as resulting from the application of Möbius' map to the method of Sebastiao e Sylva. In fact, we have that ϕ_i is associated to ψ_i by the mapping defined in Section 2, and

$$\psi_i(x) = z(\phi_i(x)) \bmod p(x), \qquad \text{for all } i, \tag{11}$$

if we choose the initial polynomials $\phi_0(x)$ and $\psi_0(x)$ such that $\psi_0(x) = z(\phi_0(x)) \bmod p(x)$. To see this, assume that (11) holds for some $i \geq 0$; we have

$$\begin{aligned}
\psi_{i+1}(x) &= J(\psi_i(x)) \bmod p(x) && \text{by definition} \\
&= J(z(\phi_i(x)) \bmod p(x) && \text{by induction assumption} \\
&= z(S(\phi_i(x))) \bmod p(x) && \text{by (7)} \\
&= z(\phi_{i+1}(x)) \bmod p(x) && \text{by definition}
\end{aligned}$$

which proves that (11) holds for all i.

Observe also that, if $p(x) = \prod_{i=1}^{n}(x - \xi_i)$, then $\phi_i(\xi_j) = \phi_0(\xi_j)^{2^i} = S \circ S \circ \cdots \circ S(\phi_0(\xi_j))$, whereas $\psi_i(\eta_j) = J \circ J \circ \cdots \circ J(\psi_0(\eta_j))$, where the functional composition $\circ$ is applied i times.

The algorithms can be implemented in the polynomial setting with fast algorithms for polynomial computation [7] or in the matrix setting where the computation modulo $p(x)$ automatically results from the application of the square function $S(\cdot)$ or Joukowski's function $J(\cdot)$ to the Frobenius matrix F. In fact, we have $\phi_i(F) = \phi_{i-1}(F)^2 = \phi_0(F)^{2^i}$, and each step of the repeated squaring iteration costs just one matrix multiplication. We also have $\psi_i(F) = J(\psi_{i-1}(F)) = J \circ \cdots \circ J(\psi_0(F))$ and the cost of each iteration with Joukowski's function is dominated by the cost of one matrix inversion.

Thus, it might seem preferable to apply Sebastiao e Sylva's iterations than Cardinal's iterations since squaring a polynomial modulo $p(x)$ is less expensive than computing the reciprocal modulo $p(x)$. However, to apply the Euclidean scheme to $p(x)$ and $\phi_i(x)$ is a numerically ill-conditioned problem. In order to overcome this difficulty, one might consider starting the computation in the domain of the unit circle, compute $F_i = F^{2^i}$ for some large enough i by means of repeated squaring, then switch to the domain of the imaginary axis by computing $H_i = (I - F_i)^{-1}(I + F_i)$ and complete the computation there. Unfortunately, the condition number of the matrix $(I - F_i)$ which must be inverted for the computation of H_i grows exponentially with i and this makes the mixed approach numerically unstable as well. Therefore, Cardinal's algorithm, or the equivalent matrix sign iteration, is the more appropriate for polynomial factorization.

This is a nice example of how the application of (2) and (3) provides algorithmic improvements for solving certain polynomial and matrix computations.

Remark 3 If we use $w(x)$ to map the coordinates of the unit circle onto the coordinates of the imaginary axis, then we are lead through (8) to the following relation between Sebastiao e Sylva's and Cardinal's algorithm: $\psi_i(x) = (-1)^i w(\phi_i(x)) \bmod p(x)$ instead of (11).

Graeffe and Chebyshev One may also apply the mapping (2) to the *variable* of a polynomial; this provides us with a different type of association between methods. Take Graeffe's iterative method[1] [26, 16] where a sequence of polynomials is defined as follows:

$$p_{i+1}(x^2) = (-1)^n p_i(x) p_i(-x), \quad p_0(x) = p(x). \tag{12}$$

Denote by $\xi_1, \ldots, \xi_n$ the roots of $p(x)$. The roots of $p_i(x)$ are $\xi_1^{2^i}, \ldots, \xi_n^{2^i}$. If k roots, $\xi_1, \ldots, \xi_k$ say, have modulus less than 1, and $n-k$ roots have modulus greater than 1, then k roots of the sequence $\{p_i(x)\}$ tend to zero and $n-k$ roots tend to infinity, and the convergence is doubly exponential.

This is the approach used in [8, 27, 28, 31] to factor the polynomial $p(x)$ with respect to the unit circle, i.e., to approximate the coefficients of the polynomials $\prod_{i=1}^{k}(x-\xi_i)$ and $\prod_{i=k+1}^{n}(x-\xi_i)$. Using the FFT-based fast polynomial arithmetic, the computation of the coefficients of $p_{i+1}(x)$, given those of $p_i(x)$, costs $O(n \log n)$ arithmetic operations.

Let us consider first, for simplicity, the relation

$$u(x^2) = (-1)^n v(x) v(-x) \tag{13}$$

where $u(x)$ and $v(x)$ are polynomials of degree n. If we replace x with $w(t)$ of (3), multiply both sides of the above equation by $(1+t)^n(1+t^{-1})^n$ and apply the transformation (4), we find from (6) that

$$(1+t)^n(1+t^{-1})^n u(w^2(t)) = (-1)^n V(t) V(1/t),$$

where $V(t) = \mathcal{P}(v(x))$. Now, $w^2(t) = w(J(t))$ by (7), so that $u(w^2(t)) = u(w(J(t)))$ and $(1+J(t))^n u(w(J(t))) = U(J(t))$, where $U(t) = \mathcal{P}(u(x))$. Since $(1+t)(1+t^{-1})/(1+J(t)) = 2$, we obtain that

$$U(J(t)) = (-2)^{-n} V(t) V(1/t). \tag{14}$$

By applying the argument above to (12), we obtain the iteration

$$q_{i+1}(J(t)) = (-2)^{-n} q_i(t) q_i(t^{-1}) \tag{15}$$

[1] first discovered by Dandelin and Lobachevski [26]

and the polynomials in that sequence are such that

$$q_i(t) = \mathcal{P}(p_i(x)) \qquad \text{for all } i.$$

If we denote by $\mu_1^{(i)}, \ldots, \mu_n^{(i)}$ the roots of $u_i(w)$, then $\mu_j^{(i+1)} = J(\mu_j^{(i)})$, $j = 1, \ldots, n$, and the sequence $\{\mu_j^{(i)}\}_i$ converges doubly exponentially to 1 or to -1 according to the sign of the real part of μ_j.

The iteration (15), introduced in [8] and called *Chebyshev's iteration* there, can be used to factor a polynomial with respect to the imaginary axis. The computation of the coefficients of $u_{i+1}(t)$, given the coefficients of $u_i(t)$ can be performed in $O(n \log^2 n)$ arithmetic operation; the higher cost (compared to the cost of Graeffe's iteration) results from the need to apply the evaluation/interpolation technique at the nodes $\cos \frac{\pi i}{2n}$, $i = 1, \ldots, n$. See [8] for more details.

In Section 5 we will show how the transition from the domain of the unit circle to the domain of the imaginary axis can be performed also in the case of matrix polynomials, and use this fact to devise new iterative methods for the solution of the matrix equation (9). Before doing so, we recall in the next section two classical technique for solving (1).

Remark 4 If we had used $w(x)$ instead of $z(x)$ to map from the unit circle to the imaginary axis, by replacing $\mathcal{P}$ of (4) with $\mathcal{P}(p(x)) = (1-t)^n p(z(t))$, we would have obtained the following equation

$$U_R(J(t)) = 2^{-n} V(t) V(1/t).$$

in place of (14), where $U_R(t) = t^n U(t^{-1})$ is the polynomial obtained by *reversing* the coefficients of $U(t)$. In this case the iteration (15) would have been replaced by the more complex

$$\begin{aligned} r_{i+1}(J(t)) &= 2^{-n} q_i(t) q_i(t^{-1}) \\ q_{i+1}(t) &= t^n r_{i+1}(t^{-1}) \end{aligned}$$

The iteration above has the same convergence properties and the same computational cost as (15).

4 Past Algorithms Revisited

Cyclic Reduction and Logarithmic Reduction operate in the domain of the unit circle and can be viewed as two different ways of extending Graeffe's iteration to matrix polynomials, while the matrix sign iteration operates in

the domain of the imaginary axis and can be viewed as an extension of Cardinal's algorithm.

Before dealing with the general case, we describe these techniques for positive recurrent QBDs, for which $n = 2$ and (1) reduces to

$$A_0 + A_1 X + A_2 X^2 = 0. \tag{16}$$

The minimal solution G of (16) has spectral radius equal to 1. Furthermore, the roots $\xi_1, \ldots, \xi_{2m}$ of the polynomial $\det(A_0 + A_1 x + A_2 x^2)$ are such that $|\xi_1| \leq \cdots \leq |\xi_{2m}|$, $\xi_m = 1 < |\xi_{m+1}|$ and ξ_m is simple (here we assume zeros at infinity if A_2 is singular).

Logarithmic Reduction We express in polynomial form the algorithm LR of [19] for the solution of (16). Let $r_i(x)$ be matrix polynomials recursively defined by

$$\begin{array}{l} r_i(z) = E_{0,i} + E_{1,i} x + E_{2,i} x^2 \\ r_{i+1}(x^2) = -\widehat{r}_i(x)\widehat{r}_i(-x), \\ \widehat{r}_i(x) = E_{1,i}^{-1} r_i(x), \end{array} \tag{17}$$

where $r_0(x) = A_0 + A_1 x + A_2 x^2$. The matrix coefficients of $r_i(x)$ and $r_{i+1}(x)$ are related by the equations

$$\begin{array}{rcl} E_{0,i+1} & = & -(E_{1,i}^{-1} E_{0,i})^2, \\ E_{2,i+1} & = & -(E_{1,i}^{-1} E_{2,i})^2, \\ E_{1,i+1} & = & I - (E_{1,i}^{-1} E_{0,i})(E_{1,i}^{-1} E_{2,i}) - (E_{1,i}^{-1} E_{2,i})(E_{1,i}^{-1} E_{0,i}) \end{array} \tag{18}$$

and the following result is proved in [19].

Theorem 1 *The minimal solution G of (16) is such that $G = G_i + O(\sigma^{2^i})$ with $\sigma = |\xi_m|/|\xi_{m+1}| < 1$, where $G_i = \sum_{j=0}^{i} \left(\prod_{k=0}^{j-1} D_{2,k} \right) D_{0,k}$, $D_{j,k} = -E_{1,i}^{-1} E_{j,i}$, $j = 0, 2$.*

Observe that the computation of (18) can be performed with one matrix inversion and 6 matrix multiplications (we do not count matrix additions since they have a lower cost $O(m^2)$ instead of $O(m^3)$). In Markov chains applications, the matrices $E_{1,i}$ are nonsingular M-matrices and their inversion can be performed in a numerically stable way without pivoting by applying Gaussian elimination with the GTH trick of [14]. Actually, these matrices are not explicitly inverted: their LU factorization is computed instead. In this way, multiplying a matrix by $U^{-1}L^{-1}$ corresponds to solving $2m$ triangular

systems and can be performed in $2m^3 + O(m^2)$ arithmetic operations. The cost of the LU factorization is dominated by $(2/3)m^3$ ops, while the cost of matrix multiplication is dominated by $2m^3$ ops. The overall cost, including two matrix multiplications needed to update G_i, is therefore $(50/3)m^3 + O(m^2)$ ops per iteration. Besides the numerically stable inversion of $E_{1,i}$, the remaining operations involve multiplications and additions of nonnegative numbers, so that there is no possibility of cancellation errors.

Cyclic Reduction Define the polynomials $p_i(x) = A_{0,i} + A_{1,i}x + A_{2,i}x^2$ by

$$p_{i+1}(x^2) = -p_i(x)A_{1,i}^{-1}p_i(-x), \tag{19}$$

with $p_0(x) = A_0 + A_1x + A_2x^2$. The coefficient matrices $A_{j,i+1}$, themselves are iteratively defined by

$$\begin{aligned} A_{1,i+1} &= A_{1,i} - A_{0,i}A_{1,i}^{-1}A_{2,i} - A_{2,i}A_{1,i}^{-1}A_{0,i}, \\ A_{0,i+1} &= -A_{0,i}A_{1,i}^{-1}A_{0,i}, \\ A_{2,i+1} &= -A_{2,i}A_{1,i}^{-1}A_{2,i} \end{aligned} \tag{20}$$

and the following property is proved in [4]:

Theorem 2 *The minimal solution G of (16) is such that $G = G_i + O(\sigma^{2^i})$, with $\sigma = |\xi_m|/|\xi_{m+1}| < 1$, where $G_i = -\widehat{A}_i^{-1}A_0$ and*

$$\widehat{A}_{i+1} = \widehat{A}_i - A_{2,i}A_{1,i}^{-1}A_{0,i}, \tag{21}$$

for $i \geq 0$, with $\widehat{A}_0 = A_1$.

The computational cost per iteration is one matrix inversion and 6 matrix multiplications, including the cost of updating $\widehat{A}_i$. The overall cost is therefore $(38/3)m^3 + O(m^2)$ ops. The same numerical stability properties hold as for LR. As a matter of fact, the two algorithms are directly related as the following theorem shows (the proof by induction is immediate and omitted).

Theorem 3 *The blocks (18, 20) generated by LR and CR are such that $E_{j,i} = A_{1,i-1}^{-1}A_{j,i}$, for $j = 0, 1, 2$ and all $i \geq 1$.*

Matrix sign function In the QBD case, (9) becomes

$$U_0 + U_1W + U_2W^2 = 0, \tag{22}$$

obtained by replacing X in (16) with $X = (I - W)^{-1}(I + W)$. Observe that it is the function $w(t)$ which is used for mapping the coordinates of the unit circle onto the coordinates of the imaginary axis. The coefficients are

$$\begin{aligned} U_0 &= A_0 + A_1 + A_2 \\ U_1 &= 2A_2 - 2A_0 \\ U_2 &= A_0 - A_1 + A_2 \end{aligned}$$

and U_2 is nonsingular. If $\xi_m = 1$ and $|\xi_{m-1}| < 1$, then the solution W of (22) which corresponds to the minimal solution G of (16), has only one null eigenvalue. Because of that null eigenvalue, one may not directly apply Joukowski's function to W. This difficulty is solved as follows.

The block Frobenius matrix

$$H = \begin{bmatrix} 0 & I \\ -U_2^{-1}U_0 & -U_2^{-1}U_1 \end{bmatrix}$$

has for eigenvalues the roots η_i, $i = 1, \ldots, 2m$, of $\det(U_0 + U_1 t + U_2 t^2)$, and $\eta_i = w(\xi_i)$, so that H has $m - 1$ eigenvalues in $\mathbb{C}^-$, one equal to zero, and m in $\mathbb{C}^+$.

If $\boldsymbol{x}$ and $\boldsymbol{y}$ are such that $U_0\boldsymbol{x} = 0$, $\boldsymbol{y}^T U_0 = 0$, $\boldsymbol{y}^T U_1 \boldsymbol{x} = 1$, then the matrix

$$\widehat{H} = H - \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^T U_1 & \boldsymbol{y}^T U_2 \end{bmatrix}$$

has m eigenvalues in $\mathbb{C}^-$ and m eigenvalues in $\mathbb{C}^+$. Therefore, the MSF iteration can be applied to $\widehat{H}$ and yields the sequence

$$\widehat{H}_{i+1} = \frac{1}{2}(\widehat{H}_i + \widehat{H}_i^{-1}), \tag{23}$$

which converges to $\lim_i \widehat{H}_i = K$. That limit may be expressed as $K = TST^{-1}$, where $S = \text{diag}\,(s_1, \ldots, s_{2m})$, $s_i = \text{sign}\,(\text{Re}\,(\eta_i))$, and $J = T^{-1}\widehat{H}T$ is the Jordan normal form of $\widehat{H}$. Now, it is possible to recover W from K as follows: by means of the QR factorization, say, of $K - I$, compute the $(2m) \times m$ matrix K^* made up by m linearly independent columns of $K - I$; partition K^* into two $m \times m$ blocks K_1^* and K_2^*. The matrix $W = (K_1^* + K_2^*)(K_1^* - K_2^*)^{-1}$ is the solution of (22) having eigenvalues in $\mathbb{C}^-$, and $G = (I + W)(I - W)^{-1} = -K_1^*(K_2^*)^{-1}$ is the minimal nonnegative solution of (16).

The cost of (23) amounts to inverting a $(2m) \times (2m)$ matrix, that is about $16m^3$ ops, to be compared to $50/3m^3$ for LR and to $38/3m^3$ for CR. For the final computation of G, we have to compute a QR factorization of $\widehat{H}_i - I$, for

a large enough value of i, obtain approximations of K_1^* and K_2^*, and compute $G = -K_1^*(K_2^*)^{-1}$ with one matrix inversion and one matrix product. The cost of the latter computation amounts to $\frac{4}{3}(2m)^3 + 6m^3$ ops.

The convergence speed of $\widehat{H}_i$ to K depends on how fast the values $J^{(i)}(\eta_j)$ converge to ± 1, where $J^{(i)}$ denotes the composition of Joukowski's function i times with itself. Defining $w_{i+1} = J(w_i)$, one has that $w_{i+1} \pm 1 = \frac{(w_i \pm 1)^2}{2w_i}$, so that one may expect slow convergence if w_0 is very close to zero or if w_0 is very large in modulus or if w_0 is very close to the imaginary axis. In the domain of the unit circle, the three conditions correspond to eigenvalues of G being close to 1, to -1 or to the unit circle. By contrast, the convergence of CR and LR is not slowed down if there are eigenvalues of modulus close to 1 provided that the ratio $|\xi_m/\xi_{m+1}|$ is sufficiently small. Therefore we may expect LR and CR to converge faster than MSF in general.

Moreover, if some value η_j is close to the imaginary axis, the matrices $\widehat{H}_i$ generated by MSF may have a very large condition number and the computation can be affected by large rounding errors [2].

Arbitrary matrix power series If $p(x) = \sum_{i=0}^{\infty} A_i x^i$ is a matrix power series or a matrix polynomial of degree n (when $A_{n+1} = A_{n+2} = \cdots = 0$), then LR and CR still apply. For CR, the equation (19) must be adjusted in the following way [4, 3]:

$$p_{i+1}(x^2) = -p_i(x)K_i(x)^{-1}p_i(-x) \tag{24}$$

where

$$K_i(x^2) = (p_i(x) - p_i(-x))/(2x)$$

with $p_0(x) = \sum_{j=0}^{+\infty} A_j x^j$.

In this generalization the quadratic matrix polynomials are replaced by the matrix power series $p_i(x)$ and the matrices $\widehat{A}_i$ of Theorem 2 are replaced by the matrix power series $\widehat{p}_i(x)$ defined below (for more details we refer the reader to [4, 3])

$$\widehat{p}_{i+1}(x) = \widehat{p}_i^{(odd)}(x) - p_i^{(even)}(x)K_i(x)^{-1}\widehat{p}_i^{(even)}(x) \tag{25}$$

where $\widehat{p}_i(x) = \sum_{j=0}^{+\infty} \widehat{A}_{j+1,i}x^j$, $\widehat{p}_0(x) = \sum_{j=0}^{+\infty} A_{j+1}x^j$, $p_i^{(even)}(x) = \sum_{j=0}^{+\infty} A_{2j,i}x^j$, $\widehat{p}_i^{(even)}(x) = \sum_{j=0}^{+\infty} \widehat{A}_{2(j+1),i}x^j$, $\widehat{p}_i^{(odd)}(x) = \sum_{j=0}^{+\infty} \widehat{A}_{2j+1,i}x^j$.

The matrix power series in the sequences $\{p_i(x)\}$ and $\{\widehat{p}_i(x)\}$ converge in the unit disk, and for this reason can be easily approximated with polynomials of suitable degree n_i (we call these the numerical degrees of the power

series). Moreover,

$$A_0 + \sum_{j=0}^{\infty} \widehat{A}_{j+1,i} G^{1+j2^i} = 0 \tag{26}$$

for all i, and the following result holds:

Theorem 4 *Let the matrix power series $p(x) = \sum_{i=0}^{+\infty} A_i x^i$ be such that $I + A_1 \geq 0$, $A_i \geq 0$, $i \neq 1$, $B = I + \sum_{i=0}^{+\infty} A_i$ is stochastic and irreducible. Assume that $p(x)$ is meromorphic in the complex plane, that $\sum_{i=1}^{+\infty} iA_i < +\infty$ and that the dominating left eigenvector $\boldsymbol{b}$ of B, normalized by $\boldsymbol{b}^T\boldsymbol{e} = 1$, where $\boldsymbol{e} = (1, \ldots, 1)^T$, satisfies $\boldsymbol{b}^T \sum_{i=1}^{+\infty} iA_i\boldsymbol{e} < 1$. Then the equation (1) has one nonnegative solution G with spectral radius 1 and $G = -(\widehat{A}_{1,i})^{-1}A_0 + O(\sigma^{-2^i})$, where $\widehat{A}_{1,i} = \widehat{p}_i(0)$ and $\sigma = \min\{|x| : \ x \in \mathbb{C}, \det p(x) = 0, \ |x| > 1\}$.*

Proof. From the hypotheses it follows that there exists a sequence of positive vectors $\boldsymbol{\pi}_j \in \mathbb{R}^m$, $j = 0, 1, 2, \ldots$, such that $\boldsymbol{\pi}_0^T(A_0 + A_1) + \boldsymbol{\pi}_1^T A_0 = 0$, $\sum_{j=0}^{i} \boldsymbol{\pi}_j^T A_{i-j} = 0$, $i = 2, 3, \ldots$, $\sum_{j=0}^{\infty} \boldsymbol{\pi}_j^T \boldsymbol{e} = 1$ (see [25]), and $\boldsymbol{\pi}_j = O(\sigma^{-j})$ (see [13]). By following the same argument as in the proof of [4, Theorem 4.1], we find that $\boldsymbol{\pi}_0^T \widehat{A}_{j,i} + \boldsymbol{v}_{j,i} = \boldsymbol{\pi}_{(j-1)2^i}$ for $j \geq 2$, where $\boldsymbol{v}_{j,i} = \sum_{k=1}^{j} \boldsymbol{\pi}_{k2^i}^T A_{j-k,i} + \boldsymbol{\pi}_{(j-1)2^i}$. Since the vector $\boldsymbol{v}_{j,i}$ is nonnegative (see [4]), $\boldsymbol{\pi}_0$ is positive and $\widehat{A}_{j,i}$ is nonnegative for $j \geq 2$, then we deduce that $\widehat{A}_{j,i} = O(\sigma^{-(j-1)2^i})$, for $j = 2, 3, \ldots$. The thesis follows from (26). □

A consequence of this theorem is that the numerical degrees n_i are bounded from above by a constant N. The computation of the $n_i + 1$ coefficients of $p_i(x)$ can be performed by means of evaluation/interpolation at the roots of 1 at a cost of $O(Nm^3 + Nm^2 \log m)$ ops; in practice, the values of n_i are roughly halved at each step of CR, which makes CR an effective tool for the numerical solution of (1) even for matrix power series. See [5] for details.

It is clear that one may similarly generalize the LR algorithm to the case of matrix power series.

Two special cases are worth mentioning. The first one is when $p(x)$ is a matrix polynomial of degree n. Then we may in principle avoid to deal with the general form (25) of the iterative procedure by considering the "re-blocked" matrix equation

$$\mathcal{A}_0 + \mathcal{A}_1\mathcal{X} + \mathcal{A}_2\mathcal{X}^2 = 0 \tag{27}$$

where $\mathcal{A}_i$ are $(n-1) \times (n-1)$ block matrices with $m \times m$ blocks and: $\mathcal{A}_2$ is the block lower triangular Toeplitz matrix with A_{n-i+j} in position (i, j) for $i \geq j$; $\mathcal{A}_1$ is the block Hessenberg block Toeplitz matrix with A_{j-i+1}

in position (i,j) for $j \geq i-1$; $\mathcal{A}_0$ is the block matrix having null blocks everywhere except for A_0 in the upper rightmost corner. We might apply to (27) the CR algorithm (20) or the LR algorithm (18); by following the same technique as in [6], it can be proved that the cost of each iteration is $O(m^3 n \log^2 m)$ ops. We see that the approach based on power series is generally much more convenient since the numerical degrees of the matrix power series involved in the computation rapidly decay.

The second case is when the matrix power series $p(x) = \sum_{i=0}^{+\infty} A_i x^i$ is such that $x + p(x)$ is a rational function, that is, when $x + p(x) = d(x)^{-1} e(x)$, where $d(x)$, $e(x)$ are matrix polynomials. Then the equation $p(X) = 0$ can be rewritten as $X = d^{-1}(X)e(X)$, or $d(X)X - e(X) = 0$. In this way we reduce the original equation to a polynomial equation and CR can be applied in its polynomial form.

The MSF algorithm can be easily applied to the case of polynomials of any degree n. The cost of each step is dominated by the inversion of an $nm \times nm$ matrix in the algebra generated by the block Frobenius matrix associated with the matrix polynomial $p(x) = \sum_{i=0}^{n} A_i x^i$. This is a Toeplitz-like matrix and its inversion costs $O(m^3 n \log^2 n)$ ops. The evaluation of the invariant subspaces by means of the SVD costs $O((nm)^3)$ ops and this is the most expensive part of the overall computation for large n.

5 Chebyshev's Iteration

We extend here Chebyshev's iteration (15) in the CR fashion. We give all details for matrix polynomials of degree $n = 2$. The case of polynomials of arbitrary degree is not given here as it is dealt with in a similar manner, starting from the general CR iteration (24).

In order to solve (16) in the domain of the unit circle, we apply the operator $\mathcal{P}$ of (4) to the polynomial $p_0(x)$ and solve the equation

$$Q_0 + Q_1 T + Q_2 T^2 = 0 \tag{28}$$

in the domain of the imaginary axis, where

$$\begin{aligned} Q_0 &= A_0 - A_1 + A_2, \\ Q_1 &= 2(A_0 - A_2), \\ Q_2 &= A_0 + A_1 + A_2. \end{aligned}$$

If T is the solution of (28) with all eigenvalues having nonnegative real parts, then $G = (T - I)(T + I)^{-1}$ is the solution of (15) with all eigenvalues in the unit circle.

We define the polynomial sequence

$$q_i(t) = (1+t)^2 p_i(w(t)) \equiv Q_{0,i} + Q_{1,i}t + Q_{2,i}t^2 \tag{29}$$

and, in order to overcome the non-commutativity of matrix multiplication, we need to find matrices K_i with the same role of $-A_{1,i}$ in (19) such that (15) generalizes to $q_{i+1}(J(t)) = q_i(t)Mq_i(t^{-1})$ for some matrix M.

First, we observe that $-A_{1,i} = (p_i(-x) - p_i(x))/(2x)$. Setting $x = w(t)$ in the equation above, we obtain that

$$\begin{aligned} -A_{1,i} &= \frac{t+1}{2(t-1)}\left(\frac{1}{(t^{-1}+1)^2}q_i(t^{-1}) - \frac{1}{(t+1)^2}q_i(t)\right) \\ &= (Q_{0,i} - Q_{2,i})/2. \end{aligned} \tag{30}$$

Now, apply to (19) the same transformations as to (13): replace x with $w(t)$ and multiply by $(1+t)^2(1+t^{-1})^2$. This leads to

$$q_{i+1}(J(t)) = \frac{1}{2}q_i(t)K_i q_i(t^{-1}) \tag{31}$$

where

$$K_i = (Q_{0,i} - Q_{2,i})^{-1}.$$

The matrix coefficients of $q_i(t)$ are related to those of $q_{i+1}(t)$ by the equations (32) below. For the sake of simplicity in proving (32), we temporarily write Q_0, Q_1, Q_2 instead of $Q_{0,i}, Q_{1,i}, Q_{2,i}$ and K instead of K_i.

We define $\sum_{j=-2}^{2} C_j t^j = q_i(t)K_i q_i(t^{-1})$ and verify by direct inspection that

$$\begin{array}{ll} C_1 = Q_1KQ_0 + Q_2KQ_1 & C_{-1} = Q_0KQ_1 + Q_1KQ_2 \\ C_2 = Q_2KQ_0 & C_{-2} = Q_0KQ_2 \\ C_0 = Q_0KQ_0 + Q_1KQ_1 + Q_2KQ_2 & \end{array}$$

We have that $C_1 = C_{-1}$ and $C_2 = C_{-2}$ since $Q_1K(Q_0 - Q_2) = Q_1 = (Q_0 - Q_2)KQ_1$ and $Q_2KQ_0 = Q_0KQ_0 - Q_0 = Q_0KQ_2$. Therefore, we obtain that

$$\begin{aligned} \sum_{j=-2}^{2} C_j t^j &= C_0 + C_1(t + t^{-1}) + C_2(t^2 + t^{-2}) \\ &= (C_0 - 2C_2) + \frac{t+t^{-1}}{2}2C_1 + \left(\frac{t+t^{-1}}{2}\right)^2 (4C_2). \end{aligned}$$

Finally, since $C_0 - 2C_2 = Q_0 - Q_2 + Q_1 K Q_1$, the coefficient matrices $Q_{0,i}$, $Q_{1,i}$, $Q_{2,i}$, of the polynomials $q_i(t)$ are iteratively determined by the following relations

$$\begin{aligned} Q_{0,i+1} &= \frac{1}{2}(Q_{0,i} - Q_{2,i} + Q_{1,i} K_i Q_{1,i}), \\ Q_{1,i+1} &= Q_{2,i} K_i Q_{1,i} + Q_{1,i} K_i Q_{0,i}, \\ Q_{2,i+1} &= 2 Q_{2,i} K_i Q_{0,i}. \end{aligned} \tag{32}$$

In order for this algorithm to be useful, we need to express in the domain of the imaginary axis the successive approximations of G which are defined in Theorem 2. This means that we need an expression for the matrices $\widehat{A}_i$ of (21) and, unfortunately, it seems that there is no direct polynomial interpretation of these. Thus, we need to explicitly relate the coefficients of $p_i(x)$ with those of $q_i(t)$. A simple calculation shows that

$$\begin{array}{ll} Q_{0,i} = A_{0,i} - A_{1,i} + A_{2,i}, & A_{0,i} = \frac{1}{4}(Q_{0,i} + Q_{1,i} + Q_{2,i}), \\ Q_{1,i} = 2(A_{0,i} - A_{2,i}), & A_{1,i} = -\frac{1}{2}(Q_{0,i} - Q_{2,i}), \\ Q_{2,i} = A_{0,i} + A_{1,i} + A_{2,i}, & A_{2,i} = \frac{1}{4}(Q_{0,i} - Q_{1,i} + Q_{2,i}). \end{array}$$

Defining $\widehat{Q}_i$ as

$$\widehat{Q}_{i+1} = \widehat{Q}_i + (Q_{0,i} - Q_{1,i} + Q_{2,i}) K_i (Q_{0,i} + Q_{1,i} + Q_{2,i})/8 \tag{33}$$

where $\widehat{Q}_0 = -\frac{1}{2}(Q_0 - Q_2)$, we readily conclude from (21) that $\widehat{Q}_i = \widehat{A}_i$ for all i. Then, the matrix sequence $G_i = -\widehat{A}_i^{-1} A_0$ which quadratically converge to G (by Theorem 2) can be written as

$$G_i = -\frac{1}{4}\widehat{Q}_i^{-1}(Q_0 + Q_1 + Q_2). \tag{34}$$

The computational cost at each step of the iterations (32, 33) amounts to one matrix inversion and 8 matrix multiplications; this cost is comparable with the one of CR.

In conclusion of this section, we point to the fact that, should it be necessary, Chebyshev's iterative scheme may be used to approximate the solution $W = (G + I)^{-1}(G - I)$ of (22) having eigenvalues with non-positive real parts. Indeed, since $G_i = -\widehat{A}_i^{-1} A_0$ provides an approximation to G, $W_i = (G_i + I)^{-1}(G_i - I)$ provides an approximation to W. From (34), a simple calculation shows that

$$W_i = (-4\widehat{Q}_i + Q_{0,i} + Q_{1,i} + Q_{2,i})^{-1}(4\widehat{Q}_i + Q_{0,i} + Q_{1,i} + Q_{2,i}) \tag{35}$$

and the following is a direct consequence of Theorem 2.

Theorem 5 *Let the equation (22) have a solution W with eigenvalues $\eta_1, \ldots, \eta_m$ having non-positive real parts. Then $W = W_i + O(\delta^{2^i})$, where W_i is defined in (32, 33, 35) and $\delta = \left|\frac{(\eta_m+1)(\eta_{m+1}-1)}{(\eta_m-1)(\eta_{m+1}+1)}\right| < 1$.*

6 Numerical Experiments

We have implemented in Fortran 95 the three techniques CR, MSF and Chebyshev's iteration (henceforth denoted as ChI) for the solution of (16). We checked the convergence speed of the three algorithms and their numerical performances. We report in Table 1 the number of steps and the residual error for three test problems. If $\widetilde{G}$ is an approximate solution of (16) we define $||A_0 + A_1\widetilde{G} + A_2\widetilde{G}^2||_\infty$ the residual error, where $||\cdot||_\infty$ denotes the infinity norm.

Problem 1 The $m \times m$ matrices A_0, A_1, A_2, where $m = 32$, are given by, $A_0 = M_1^{-1}M_0$, $A_1 = I$, $A_2 = M_1^{-1}M_2$, with $(M_2)_{i,j} = \alpha_i$ if $j = i+1 \pmod m$, $(M_2)_{m,m} = \sigma$, $(M_2)_{i,j} = 0$ elsewhere, $(M_0)_{i,j} = \beta_i$ if $j = i-1 \pmod m$, $(M_0)_{i,j} = 0$ elsewhere, $M_1 = \text{diag}\,(\gamma_1, \ldots, \gamma_m)$, $\gamma_i = -\alpha_i - \beta_i$, $i = 1, \ldots, m-1$, $\gamma_m = -\alpha_m - \beta_m - \sigma$. The numerical values of the parameters are $\alpha_i = 1$, $\beta_i = 1.2$, for $i = 1, \ldots, 16$, $\beta_i = 0.85$, for $i = 17, \ldots, 32$, $\sigma = 0.001$. For this problem the eigenvalues are close to the unit circle on both sides: it holds $\xi_{m-1} = 0.998975$, $\xi_m = 1$ and $\xi_{m+1} = 1.00978$. In this, case, since $|\xi_m/\xi_{m+1}| < |\xi_{m-1}/\xi_m|$, CR performs better than MSF, as seen in Table 1. Moreover that condition number of $\widehat{H}$ and of K is $1.2 \cdot 10^{10}$ and $6.4 \cdot 10^6$, respectively, thus leading to instability problems for MSF, and to its much larger residual error. The algorithm ChI has the same behavior as CR, as expected.

Problem 2 This example is taken from [20, Page 208]. It represents an M/M/1 queue in a random environment where periods of severe overflows alternate with periods of low arrivals. The $m \times m$ matrices A_0, A_1, A_2, where $m = 8$, are given by, $A_0 = M_1^{-1}M_0$, $A_1 = I$, $A_2 = M_1^{-1}M_2$, with $M_2 = \rho \cdot \text{diag}\,(\alpha_1, \ldots, \alpha_m)$ $M_0 = \text{diag}\,(\beta_1, \ldots, \beta_m)$ $(M_1)_{i,j} = 1$ if $j = (i \bmod m)+1$, $(M_1)_{i,j} = -1 - \rho\alpha_i - \beta_i$ if $j = i$, $(M_1)_{i,j} = 0$ elsewhere, The numerical values of the parameters are $\boldsymbol{\alpha} = (0.2, 0.2, 0.2, 0.2, 13, 1, 1, 0.2)$, $\beta_i = 2$, for $i = 1, \ldots, m$, $\rho = 0.99$. Here, $\xi_m = 1$, $\xi_{m+1} = 1.00188$, and the remaining zeros are far from the unit circle. This is a critical case for the three algorithms, but CR and ChI converge faster to a more accurate solution than MSF. The condition number of $\widehat{H}$ and of K is $5.9 \cdot 10^8$ and $2.2 \cdot 10^6$, respectively.

	Problem 1		Problem 2		Problem 3	
	steps	residual	steps	residual	steps	residual
CR	12	6.7e-16	14	4.4e-16	29	6.7e-16
MSF	16	3.1e-10	19	2.3e-12	85	1.0e-1
ChI	12	6.3e-15	14	3.4e-16	18	4.9e-12

Table 1: Number of iterations and residual error

Problem 3 This example is taken from [23]. The matrices A_0, A_1, A_2 have all the entries equal to $\alpha = \frac{\rho-1}{3(m-1)}$ for $i \neq j$, $(A_0)_{i,i} = -\rho$, $(A_1)_{i,i} = 1$, $(A_2)_{i,i} = 0$, for $i = 1, \ldots, m$, and $m = 32$. We have chosen $\rho = 0.99$, so that $\xi_m = 1$, $\xi_{m+1} = 1.00000003$, and the remaining zeros are equal and far from the unit circle. This is a very difficult problem since ξ_m/ξ_{m+1} is very close to 1. The condition number of $\widehat{H}$ and of K is $4.2 \cdot 10^{23}$ and $1.8 \cdot 10^{16}$, respectively. The MSF is not able to approximate the solution at all, ChI is not able to reach a really small residual error, and CR still works very well, providing a very accurate result.

7 Conclusions

We considered the problem of solving polynomial matrix equations in terms of polynomial computations. We introduced a mapping of the complex plane and pointed out that it sets a relationship between Joukowski's and the square functions. This allowed us to relate different existing algorithms to each others and to devise new ones, each algorithm having its own version in the domain of the unit circle and in the domain of the imaginary axis.

In particular, we have seen that Akar and Sohrabi's algorithm is the matrix version of Cardinal's method which acts in the domain of the imaginary axis while its formulation in the domain of the unit circle corresponds to Sebastiao e Sylva's algorithm. We have also introduced Chebyshev's iterations for matrix polynomials which correspond, in the domain of the imaginary axis, to the Logarithmic Reduction and to the Cyclic Reduction algorithms.

By comparing Cyclic Reduction, Chebyshev's iteration and Akar and Sohrabi's method we have shown that for problems originated from queueing theory, Cyclic Reduction and its new version in the domain of the imaginary axis have better convergence performances and lower cost.

In fine, we pointed out that Cyclic Reduction (and Logarithmic Reduction) have better numerical properties than their counterpart (ChI) in the

domain of the imaginary axis. This points to the importance of remaining in the domain of the unit circle where we deal with probabilities: quantities which are positive and less than 1.

A Chebyshev's LR variant

If we apply the transformation (2) to (17) and define $s_i(t) = (1+t)^2 r_i(w(t)) = S_{0,i} + S_{1,i}t + S_{2,i}t^2$, we obtain $E_{1,i} = (r_i(x) - r_i(-x))/(2x) = (S_{2,i} - S_{0,i})/2$, where $r_i(x) = E_{0,i} + E_{1,i}x + E_{2,i}x^2$ are defined in (17). Moreover, we have

$$\begin{aligned} s_{i+1}(J(t)) &= -\widehat{s}_i(t)\widehat{s}_i(t^{-1}) \\ \widehat{s}_i(t) &= H_i s_i(t) \\ H_i &= 2(S_{2,i} - S_{0,i})^{-1} \end{aligned} \tag{36}$$

where the initial polynomial is $s_0(t) = (A_0 - A_1 + A_2) + 2(A_0 - A_2)t + (A_0 + A_1 + A_2)t^2 = S_0 + S_1 t + S_2 t^2$.

The coefficients of the matrix polynomials $s_i(t)$ are related as follows,

$$\begin{aligned} S_{0,i+1} &= -(4I + (H_i S_{2,i})^2) \\ S_{1,i+1} &= -2H_i(S_{1,i}H_i S_{0,i} + S_{2,i}H_i S_{1,i}) \\ S_{2,i+1} &= -4H_i S_{2,i} H_i S_{0,i}. \end{aligned} \tag{37}$$

Theorem 1 provides approximations to G with $D_{0,i} = H_i(S_{0,i} + S_{1,i} + S_{2,i})$, $D_{2,i} = H_i(S_{0,i} - S_{1,i} + S_{2,i})$.

References

[1] N. Akar and K. Sohraby. An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. *Commun. Statist. Stochastic Models*, 13:381–416, 1997.

[2] Z. Bai and J. Demmel. Using the matrix sign function to compute invariant subspaces. *SIAM J. Matrix Anal. Appl.*, 19(1):205–225 (electronic), 1998.

[3] D. A. Bini and B. Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 21–38. Kluwer Academic Publisher, 1995.

[4] D. A. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:906–926, 1996.

[5] D. A. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 15:57–74, 1997.

[6] D. A. Bini and B. Meini. Using displacement structure for solving non-skip-free M/G/1 type Markov chains. In A. Alfa and S. Chakravarthy, editors, *Advances in Matrix Analytic Methods for Stochastic Models - Proceedings of the 2nd international conference on matrix analytic methods*, pages 17–37. Notable Publications Inc, NJ, 1998.

[7] D. A. Bini and V. Pan. *Matrix and Polynomial Computations, Vol. 1: Fundamental Algorithms*. Birkhäuser, Boston, 1994.

[8] D. A. Bini and V. Y. Pan. Graeffe's, Chebyshev-like, and Cardinal's processes for splitting a polynomial into factors. *J. Complexity*, 12:492–511, 1996.

[9] Jean-Paul Cardinal. On two iterative methods for approximating the roots of a polynomial. In *The mathematics of numerical analysis (Park City, UT, 1995)*, pages 165–188. Amer. Math. Soc., Providence, RI, 1996.

[10] P. Favati and B. Meini. On functional iteration methods for solving M/G/1 type Markov chains. In A. Alfa and S. Chakravarthy, editors, *Advances in Matrix Analytic Methods for Stochastic Models - Proceedings of the 2nd international conference on matrix analytic methods*, pages 39–54. Notable Publications Inc, NJ, 1998.

[11] P. Favati and B. Meini. On functional iteration methods for solving nonlinear matrix equations arising in queueing problems. *IMA J. of Numerical Analysis*, 19:39–49, 1999.

[12] H. R. Gail, S. L. Hantler, and B. A. Taylor. Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains. *Adv. in Appl. Probab.*, 28:114–165, 1996.

[13] H. R. Gail, S. L. Hantler, and B. A. Taylor. Use of characteristics roots for solving infinite state Markov chains. In W. K. Grassmann, editor, *Computational Probability*, pages 205–255. Kluwer Academic Publishers, 2000.

[14] W. K. Grassman, M. I. Taksar, and D. P. Heyman. Regenerative analysis and steady state distribution for Markov chains. *Oper. Res.*, 33:1107–1116, 1985.

[15] N. J. Higham and H.-M. Kim. Numerical analysis of a quadratic matrix equation. Numerical Analysis Report No. 347, Manchester Centre for Computational Mathematics, Manchester, England, August 1999. To appear in IMA J. of Numerical Analysis.

[16] A. S. Householder. *The Numerical Treatment of a Single Nonlinear Equation.* International Series in Pure and Applied Mathematics. McGraw-Hill Inc., New York, NY, 1970.

[17] G. Latouche. Algorithms for infinite Markov chains with repeating columns. In C.D. Meyer and R.J. Plemmons, editors, *Linear Algebra, Queueing Models and Markov Chains*, pages 231–265. Springer-Verlag, New York, 1993.

[18] G. Latouche. Newton's iteration for non-linear equations in Markov chains. *IMA J. of Numerical Analysis*, 14:583–598, 1994.

[19] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Probability*, 30:650–674, 1993.

[20] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling.* ASA-SIAM Series on Statistics and Applied Probability 5. SIAM, 1999.

[21] G. Latouche and G.W. Stewart. Numerical methods for M/G/1 type queues. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 571–581. Kluwer Academic Publishers, 1995.

[22] B. Meini. New convergence results on functional iteration techniques for the numerical solution of M/G/1 type Markov chains. *Numer. Math.*, 78:39–58, 1997.

[23] B. Meini. Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method. *Adv. Perf. Anal.*, 1:215–225, 1998.

[24] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* The Johns Hopkins University Press, Baltimore, MD, 1981.

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

[26] M. A. Ostrowski. Recherches sur la methode de Graeffe et les zeros des polynomes et des series de Laurent. *Acta Math.*, 72:99–257, 1940.

[27] V. Y. Pan. Algebraic complexity of computing polynomial zeros. *Computers & Math. (with Applications)*, 14, 4:285–304, 1987.

[28] V. Y. Pan. Sequential and parallel complexity of approximate evaluation of polynomial zeros. *Computers & Math. (with Applications)*, 14, 8:591–622, 1987.

[29] V. Ramaswami. Nonlinear matrix equations in applied probability - Solution techniques and open problems. *SIAM Review*, 30:256–263, 1988.

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

[31] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Math. Dept., University of Tübingen, Tübingen, Germany, 1982.

[32] J. Sebastiao e Silva. Sur une méthode d'approximation semblable à celle de Gräffe. *Portugaliae Math.*, 2:271–279, 1941.

Improved Cyclic Reduction for Solving Queueing Problems

Dario Andrea Bini and Beatrice Meini
Dipartimento di Matematica
Università di Pisa
Via Buonarroti 2, 56127 Pisa, Italy
bini@dm.unipi.it, meini@dm.unipi.it

Abstract

The cyclic reduction technique [7], rephrased in functional form [3], provides a numerically stable, quadratically convergent method, for solving the matrix equation $X = \sum_{i=0}^{+\infty} X^i A_i$, where the A_i's are nonnegative $k \times k$ matrices such that $\sum_{i=0}^{+\infty} A_i$ is column stochastic. In this paper we propose a further improvement of the above method, based on a pointwise evaluation/interpolation at a suitable set of Fourier points, of the functional relations defining each step of cyclic reduction [3]. This new technique allows us to devise an algorithm based on FFT having a lower computational cost and a higher numerical stability. Numerical results and comparisons are provided.

Keywords. Queueing problems, M/G/1 type matrices, cyclic reduction, Toeplitz matrices, FFT.

1 Introduction

The mathematical modelling of many queueing problems leads to solving the infinite system

$$(I - P)\boldsymbol{\pi} = 0, ||\boldsymbol{\pi}||_1 = 1, \tag{1}$$

where P is a column stochastic matrix in block Hessenberg form having the block Toeplitz-like structure

$$P = \begin{pmatrix} B_1 & A_0 & & \bigcirc & \\ B_2 & A_1 & A_0 & & \\ B_3 & A_2 & A_1 & A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} \tag{2}$$

and the blocks A_i, B_i are nonnegative $k \times k$ matrices such that $\sum_{i=0}^{\infty} A_i$, $\sum_{i=1}^{\infty} B_i$ are column stochastic. Matrices of the structure (2) are known in literature as

stochastic matrices of M/G/1 type [16]. It is well known that the solution of (1) exists uniquely if the matrix P is irreducible and positive recurrent [8], moreover the computation of $\boldsymbol{\pi}$ can be obtained from the computation of the minimal nonnegative solution G of the nonlinear matrix equation

$$X = \sum_{i=0}^{\infty} X^i A_i, \tag{3}$$

by means of the Ramaswami formula [16], [17], [14], where X is a $k \times k$ matrix. This makes of great interest the study of efficient numerical methods for solving equation (3).

Recently, the analysis of numerical methods for the computation of the solution G of (3) has been largely developed. Classical linearly convergent methods, based on functional iteration techniques, have been studied in detail in [15], [10], [16] and [18]. New quadratically convergent methods have been proposed in [3], [13], [11], [20]. These methods show their effectiveness specially in the cases where classical algorithms based on functional iterations converge very slowly [3]. From the numerical experiments performed so far [1], [3], the most efficient method seems to be the one of [3]. Indeed, the methods proposed in [13], [11], even if quadratically convergent, have a much higher computational cost per iteration.

The method of [3] consists in rewriting the matrix equation (3) in matrix form as

$$(X, X^2, X^3, \ldots) \begin{pmatrix} I - A_1 & -A_0 & & \bigcirc & \\ -A_2 & I - A_1 & -A_0 & & \\ -A_3 & -A_2 & I - A_1 & -A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} = (A_0, O, O, \ldots) \tag{4}$$

and in applying the cyclic reduction technique, rephrased in functional form, to solve (4). A single step of cyclic reduction consists in performing an odd-even permutation of the block rows and the block columns of the matrix (4) followed by one step of block Gaussian elimination. It is shown in [3] that this transformation keeps unchanged the structure of the system. By means of a recursive application of cyclic reduction, it is generated a sequence of block Hessenberg, block Toeplitz-like infinite systems that quadratically converges to a limit system whose solution can be explicitely evaluated.

The matrix of the system at step n, having the same block Toeplitz-like, block Hessenberg structure, of (2), is fully defined by its first and second block columns. The block entries of the first block column, denoted by $I - \widehat{A}_1^{(n)}$, $-\widehat{A}_2^{(n)}$, $-\widehat{A}_3^{(n)}, \ldots$, and the block entries of the second block column, denoted by $-A_0^{(n)}$, $I - A_1^{(n)}$, $-A_2^{(n)}, \ldots$, define two formal matrix power series $\widehat{\varphi}^{(n)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{i+1}^{(n)} z^i$, $\varphi^{(n)}(z) = \sum_{i=0}^{+\infty} A_i^{(n)} z^i$ associated with the system. Moreover, an

explicit functional relation intercurs between $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$ and $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$:

$$(\varphi^{(n+1)}(z), \widehat{\varphi}^{(n+1)}(z)) = F(\varphi^{(n)}(z), \widehat{\varphi}^{(n)}(z)) \tag{5}$$

for a suitable F, which expresses in functional form the cyclic reduction step.

The functional representation (5) allows one to reduce the overall computation for performing formal operations between matrix power series, which are numerically truncated to matrix polynomials, i.e., polynomials having matrix coefficients. The use of fast polynomial arithmetic, based on FFT [4], [5], for computing products and reciprocals of polynomials, provides a tool for dramatically reducing the computational cost, arriving at an efficient and reliable implementation of the algorithm [3].

In this paper we propose an improvement of the algorithm of [3], based on a more efficient implementation of the functional relations (5) which the algorithm relies on. More precisely, the idea at the basis of our improvement consists in a pointwise evaluation of (5) at a suitable set of Fourier points. From these values we may recover the block coefficients of the matrix series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$ once we are given the block coefficients of $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$. This evaluation/interpolation procedure provides accurate approximations of the matrix power series if the number of Fourier points is large enough. In fact, we introduce a modification of the FFT algorithm which allows us to compute the values of $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$ at the even-indexed Fourier points and, if needed, to compute the values of these power series taken on at the odd-indexed Fourier points without having to recompute all the values from scratch.

In order to determine the number of Fourier points sufficient to reach the desired accuracy, we introduce a criterium allowing us to control the approximation error and to adjust dynamically the number of Fourier points at each step of the algorithm with no waste of computation.

By using this different strategy we arrive at a new version of the algorithm which, besides being much easier to implement, leads to a reduction of the computational cost of a factor of about 2 and is numerically more stable.

The paper is organized as follows. In Section 2 we recall the algorithm based on cyclic reduction and the main convergence results. In Section 3 we propose the new algorithm based on the pointwise evaluation of (5). In Section 4 we introduce the basic tools for FFT computations. In Section 5 we describe the implementation of the algorithm in Fortran 90 and present numerical results.

2 The cyclic reduction algorithm.

In this section we recall the main results concerning the cyclic reduction method applied to solve equation (3). For major details we refer the reader to [3] and [2].

The key idea consists in rewriting the nonlinear equation (3) in the matrix form (4). In order to solve the block Hessenberg, block Toeplitz infinite system

(4) we recursively apply the cyclic reduction algorithm [7], i.e. a block odd-even permutation followed by one step of Gaussian elimination, thus generating the sequence of block Hessenberg, block Toeplitz-like infinite systems

$$(X, X^{2^n+1}, X^{2\cdot 2^n+1}, \ldots) \begin{pmatrix} I-\widehat{A}_1^{(n)} & -A_0^{(n)} & & \bigcirc & \\ -\widehat{A}_2^{(n)} & I-A_1^{(n)} & -A_0^{(n)} & & \\ -\widehat{A}_3^{(n)} & -A_2^{(n)} & I-A_1^{(n)} & -A_0^{(n)} & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} =$$

$$(A_0, O, O, \ldots), \ n \geq 0,$$

$$A_0^{(0)} = A_0, \ \widehat{A}_i^{(0)} = A_i^{(0)} = A_i, \ i \geq 1. \tag{6}$$

The block entries $\{A_i^{(n+1)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(n+1)}\}_{i\geq 1}$, obtained at step $n+1$, are related to the block entries $\{A_i^{(n)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(n)}\}_{i\geq 1}$, obtained at step n, by means of functional relations, that express in compact form the block odd-even permutation and the Gaussian elimination of one step of cyclic reduction. More precisely, for any $n \geq 0$, let us associate with the matrix sequences $\{A_i^{(n)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(n)}\}_{i\geq 1}$, the formal matrix power series $\varphi^{(n)}(z) = \sum_{i=0}^{+\infty} A_i^{(n)} z^i$, $\widehat{\varphi}^{(n)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{i+1}^{(n)} z^i$, respectively. Then the following functional relations hold:

$$\begin{cases} \varphi^{(n+1)}(z) = z\varphi_{odd}^{(n)}(z) + \varphi_{even}^{(n)}(z)\left(I - \varphi_{odd}^{(n)}(z)\right)^{-1} \varphi_{even}^{(n)}(z) \\ \widehat{\varphi}^{(n+1)}(z) = \widehat{\varphi}_{odd}^{(n)}(z) + \varphi_{even}^{(n)}(z)\left(I - \varphi_{odd}^{(n)}(z)\right)^{-1} \widehat{\varphi}_{even}^{(n)}(z) \end{cases} \tag{7}$$

where

$$\begin{array}{ll} \varphi_{even}^{(n)}(z) = \sum_{i=0}^{+\infty} A_{2i}^{(n)} z^i, & \varphi_{odd}^{(n)}(z) = \sum_{i=0}^{+\infty} A_{2i+1}^{(n)} z^i \\ \widehat{\varphi}_{even}^{(n)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{2(i+1)}^{(n)} z^i, & \widehat{\varphi}_{odd}^{(n)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{2i+1}^{(n)} z^i. \end{array} \tag{8}$$

The relations (7), besides to express in compact form the recursions which the cyclic reduction method is based on, provide a basic tool for the efficient computation of the matrices $\{A_i^{(n)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(n)}\}_{i\geq 1}$, as we will explain in the next section.

The following results are fundamental for devising an efficient algorithm based on cyclic reduction:

Theorem 1 *The blocks $\{A_i^{(n)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(n)}\}_{i\geq 1}$ are nonnegative matrices such that $\sum_{i=0}^{+\infty} A_i^{(n)}$, $A_0 + \sum_{i=1}^{+\infty} \widehat{A}_i^{(n)}$ are stochastic. Moreover, if the matrix $I - \sum_{i=0}^{+\infty} G^{i\cdot 2^j} \widehat{A}_{i+1}^{(n)}$ is nonsingular, then*

$$G = A_0 \left(I - \sum_{i=0}^{+\infty} G^{i\cdot 2^j} \widehat{A}_{i+1}^{(n)} \right)^{-1}. \tag{9}$$

Under mild conditions, usually satisfied in the applications, the block entries $A_i^{(n)}$ and $\widehat{A}_i^{(n)}$ quadratically converge to zero, for $i \geq 2$, for $n \to \infty$, as stated by the following theorem [3]:

Theorem 2 *Let $G' = \lim_{n\to\infty} G^n$. Then the following convergence properties hold:*

1. $\lim_{n\to\infty} \widehat{A}_i^{(n)} = 0$, *for* $i \geq 2$;
2. *if the entries of the matrix* $(I - \sum_{i=1}^{+\infty} A_i^{(n)})^{-1}$ *are bounded above by a constant, then the sequence of matrices*

$$R^{(n)} = A_0^{(n)} \left(I - \sum_{i=1}^{+\infty} A_i^{(n)} \right)^{-1} \tag{10}$$

   converges quadratically to the matrix G';
3. *if the solution G of (3) is irreducible then*

$$\lim_{n\to\infty} A_0^{(n)} (I - A_1^{(n)})^{-1} = G'$$
$$\lim_{n\to\infty} A_i^{(n)} = 0, \ i \geq 2.$$

Under the assumptions of theorems 1 and 2 the following algorithm for the numerical computation of the matrix G can be applied.

Algorithm 2.1

1. Apply cyclic reduction to (4), by means of (7), obtaining the sequence (6) of infinite systems defined by the blocks $A_i^{(n)}$,$\widehat{A}_i^{(n)}$, $n = 1, 2, \ldots, q$, until one of the following conditions is satisfied

   (C1) $|R^{(q)} - R^{(q-1)}| < \epsilon E$,

   (C2) $\mathbf{e}^T (I - A_0^{(q)} (I - A_1^{(q)})^{-1}) < \epsilon \mathbf{e}^T$,

   (C3) $\mathbf{e}^T (I - A_0 (I - \widehat{A}_1^{(q)})^{-1}) < \epsilon \mathbf{e}^T$,

   where, at each step n, the matrix $R^{(n)}$ is defined in (10), $\epsilon > 0$ is fixed and E is the $k \times k$ matrix having all the entries equal to 1.

2. Compute an approximation of the matrix G:

   (a) if condition **(C1)** or condition **(C2)** is verified compute an approximation of G by replacing, in the right hand side of (9), for $n = q$, the positive powers of G with $R^{(q)}$;

   (b) if condition **(C3)** is verified an approximation of G is given by $A_0 (I - \widehat{A}_1^{(q)})^{-1}$.

3 Pointwise Cyclic Reduction

The effectiveness of algorithm 2.1 relies on the possibility of computing the blocks $A_i^{(n)}$, $\widehat{A}_i^{(n)}$, at each step n in an efficient way: an improved computation of such matrices can lead to a strong reduction of the computational cost of each iteration.

According to the ideas developed in [3] and [2], we may truncate the sequences $\{A_i^{(n)}\}_i$ $\{\widehat{A}_i^{(n)}\}_i$ to the index m_n and $\widehat{m}_n$, respectively, such that $\mathbf{e}^T(I - \sum_{i=0}^{m_n} A_i^{(n)}) < \sigma\mathbf{e}^T$ and $\mathbf{e}^T(I - A_0 - \sum_{i=1}^{\widehat{m}_n} \widehat{A}_i^{(n)}) < \sigma\mathbf{e}^T$, where σ is an upper bound to the machine precision of the floating point arithmetic used in the computation, and is related to the accuracy of the computed approximation of G. In other words, due to theorem 1, we consider only those blocks which make the matrices $\sum_{i=0}^{m_n} A_i^{(n)}$ and $A_0 + \sum_{i=1}^{\widehat{m}_n} \widehat{A}_i^{(n)}$ numerically stochastic with respect to σ. A nonnegative matrix A is said *numerically stochastic with respect to* σ, or σ-*stochastic*, if $\mathbf{e}^T(I - A) < \sigma\mathbf{e}^T$. In this way, the matrix series $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$ are replaced by matrix polynomials of degree m_n, $\widehat{m}_n - 1$, respectively, that we denote with the same symbols $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$. The degrees m_n and $\widehat{m}_n - 1$ are called the *numerical degrees* of the series $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$, respectively. Thus, relations (7) can be rewritten in terms of matrix polynomials, in the variable z, instead of matrix power series, and all the computations can be performed by means of techniques based on Fast Fourier Transform, drastically reducing the computational cost.

In [2] and [3] we have extended to matrix polynomials the FFT-based techniques which have been devised for scalar polynomial arithmetic [5], [4], and we have reduced all the operations involved in (7) to multiply matrix polynomials and invert a matrix polynomial modulo z^m, for a suitable m (a detailed description of the algorithms is provided in [3]). In this way the computational cost to perform one step of cyclic reduction is reduced to $O(k^3 M_n + k^2 M_n \log M_n)$ arithmetic operations, where M_n is such that the matrices $\sum_{i=0}^{M_n} A_i^{(n)}$ and $A_0 + \sum_{i=1}^{M_n} \widehat{A}_i^{(n)}$ are numerically stochastic, versus the cost of $O(k^3 M_n^2)$ if the customary arithmetic were used.

In this section we propose a new improved method for the computation of one step of cyclic reduction, based on a point-wise evaluation at the roots of 1, of the series involved in (7). The new approach consists in performing the following steps:

1. Evaluating the series $\varphi_{odd}^{(n)}(z)$, $\varphi_{even}^{(n)}(z)$, $\widehat{\varphi}_{odd}^{(n)}(z)$, $\widehat{\varphi}_{even}^{(n)}(z)$ at the M_n-th roots of 1, where $M_n - 1 = 2^p - 1$ is an upper bound to the numerical degree of the above series.

2. Performing a pointwise evaluation of (7) at the M_n-th roots of 1.

3. Computing the coefficients of the matrix polynomials $P(z)$ and $\widehat{P}(z)$ of

degree $M_n - 1$ which interpolate the values of the matrix series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$ obtained at stage 2.

4. Checking whether or not the matrix polynomials $P(z)$ and $\widehat{P}(z)$ are good approximations of the series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$, respectively.

5. If the polynomials $P(z)$ and $\widehat{P}(z)$ are poor approximations of $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$, set $M_n = 2M_n$ and repeat steps 1–5 until the accuracy of the approximation is reached.

Due to the properties of FFT, in each doubling step, part of the results is already available at no cost. Moreover, unlike in the implementation of the algorithm of [2] and [3], the computation of the reciprocal of a polynomial is avoided and the order of the involved DFT and IDFT is kept to its minimum value, thus substantially reducing the computational cost per iteration.

The following properties of cyclic reduction and of FFT provide a good test to check the accuracy of the approximations $P(z)$, $\widehat{P}(z)$ of the series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$ needed at step 4:

Theorem 3 *Let, for any $n \geq 0$,*

$$\boldsymbol{\alpha}^{(n)T} = \mathbf{e}^T \sum_{i=1}^{\infty} i A_i^{(n)}, \ \widehat{\boldsymbol{\alpha}}^{(n)T} = \mathbf{e}^T \sum_{i=1}^{\infty} i \widehat{A}_{i+1}^{(n)}.$$

Then the following recursive relations hold:

$$\begin{aligned} \boldsymbol{\alpha}^{(n+1)T} &= \left(\mathbf{e}^T + \boldsymbol{\alpha}^{(n)T} - (\mathbf{e}^T - \boldsymbol{\alpha}^{(n)T})(I - \varphi_{odd}^{(n)}(1))^{-1}\varphi_{even}^{(n)}(1)\right)/2 \\ \widehat{\boldsymbol{\alpha}}^{(n+1)T} &= \left(\widehat{\boldsymbol{\alpha}}^{(n)T} - (\mathbf{e}^T - \boldsymbol{\alpha}^{(n)T})(I - \varphi_{odd}^{(n)}(1))^{-1}\widehat{\varphi}_{even}^{(n)}(1)\right)/2. \end{aligned} \tag{11}$$

Proof. Observe that, at each step n, the following relation holds:

$$(\mathbf{0}, \mathbf{e}^T, 2\mathbf{e}^T, 3\mathbf{e}^T, \ldots) \begin{pmatrix} I - \widehat{A}_1^{(n)} & -A_0^{(n)} & & \bigcirc & \\ -\widehat{A}_2^{(n)} & I - A_1^{(n)} & -A_0^{(n)} & & \\ -\widehat{A}_3^{(n)} & -A_2^{(n)} & I - A_1^{(n)} & -A_0^{(n)} & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix} =$$

$$(-\widehat{\boldsymbol{\alpha}}^{(n)T}, \mathbf{e}^T - \boldsymbol{\alpha}^{(n)T}, \mathbf{e}^T - \boldsymbol{\alpha}^{(n)T}, \mathbf{e}^T - \boldsymbol{\alpha}^{(n)T}, \ldots).$$

By applying to the above system one step of cyclic reduction we easily obtain formulae (11).

Proposition 4 *Let $s(z) = \sum_{i=0}^{\infty} s_i z^i$ be a series converging in the closed unit disk. Let $r^{(m)}(z) = \sum_{i=0}^{m-1} r_i z^i$ be the polynomial of degree $m-1$ interpolating*

the series $s(z)$ at the m-th roots of 1. Then, for $i = 0, \ldots, m-1$, the coefficients r_i are given by

$$r_i = \sum_{j=0}^{\infty} s_{i+jm}.$$

Proof Let ω be a principal m-th root of 1. Then the coefficient r_i, for $i = 0, \ldots, m-1$, is given by

$$r_i = \frac{1}{m} \sum_{h=0}^{m-1} \omega^{-ih} s(\omega^h) = \frac{1}{m} \sum_{h=0}^{m-1} \sum_{j=0}^{\infty} \omega^{(-i+j)h} s_j = \sum_{j=0}^{\infty} s_{i+jm}.$$

The above proposition can be easily extended to the case of matrix power series, thus leading to the following

Theorem 5 *Let, at step n, $P^{(m)}(z) = \sum_{i=0}^{m-1} P_i z^i$ and $\widehat{P}^{(m)} = \sum_{i=0}^{m-1} \widehat{P}_i z^i$ be the matrix polynomials of degree $m-1$ interpolating the series $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$ at the m-th roots of 1. Then the following inequalities hold:*

$$\mathbf{e}^T \textstyle\sum_{i=m}^{\infty} A_i^{(n)} \leq \boldsymbol{\alpha}^{(n)T} - \mathbf{e}^T \sum_{i=1}^{m-1} iP_i$$

$$\mathbf{e}^T \textstyle\sum_{i=m+1}^{\infty} \widehat{A}_i^{(n)} \leq \widehat{\boldsymbol{\alpha}}^{(n)T} - \mathbf{e}^T \sum_{i=1}^{m-1} i\widehat{P}_i.$$

Proof The proof readily follows from Proposition 4 and from the definition of $\boldsymbol{\alpha}^{(n)}$ and $\widehat{\boldsymbol{\alpha}}^{(n)}$ given in theorem 3.

Theorems 3 and 5 provide a good test to check the accuracy of the approximations of the series $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$ at each step n. Indeed, suppose to know the coefficients of the series $\varphi^{(n)}(z)$, $\widehat{\varphi}^{(n)}(z)$, and to compute the coefficients P_i, $\widehat{P}_i$, $i = 0, \ldots, m-1$, of the approximations $P^{(m)}(z) = \sum_{i=0}^{m-1} P_i z^i$, $\widehat{P}^{(m)}(z) = \sum_{i=0}^{m-1} \widehat{P}_i z^i$ of the series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$ by interpolating the functional relations (7) at the m-th roots of 1. From Theorem 5 it follows that, if

$$\boldsymbol{\alpha}^{(n+1)T} - \mathbf{e}^T \sum_{i=1}^{m-1} iP_i \leq \epsilon \mathbf{e}^T \tag{12}$$

and

$$\widehat{\boldsymbol{\alpha}}^{(n+1)T} - \mathbf{e}^T \sum_{i=1}^{m-1} i\widehat{P}_i \leq \epsilon \mathbf{e}^T, \tag{13}$$

then the matrices $\sum_{i=0}^{m-1} A_i^{(n+1)}$ and $A_0 + \sum_{i=1}^{m} \widehat{A}_i^{(n+1)}$ are ϵ–stochastic. Whence, for ϵ "small enough", the series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$, can be truncated to polynomials of degree $m-1$, and $P^{(m)}(z)$, $\widehat{P}^{(m)}(z)$ are good approximations. It is important to point out that (12) and (13) can be easily applied without knowing the coefficients of the series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$. In fact, $\boldsymbol{\alpha}^{(n+1)}$ and $\widehat{\boldsymbol{\alpha}}^{(n+1)}$ can be explicitely obtained by means of (11) at the cost of $O(k^3)$ arithmetic operations. This provides an efficient tool to check the accuracy of the approximations $P^{(m)}(z)$, $\widehat{P}^{(m)}(z)$ to $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$.

4 Computing DFT's.

The efficient computation of the functions (7) relies on the possibility of evaluating/interpolating a matrix power series, which is numerically reduced to a matrix polynomial, at the roots of 1 by means of FFT. In this section we introduce and analyze some tools and computations needed for dealing with this problem. Firstly, we recall well-known formulae for the computation of DFT's and IDFT's of real vectors [6], [9]. Then we provide suitable formulae for similar computations where part of the result is already available at no cost.

Throughout this section $\mathbf{i}$ is the imaginary unit such that $\mathbf{i}^2 = -1$, $\omega_m = \cos\frac{2\pi}{m} + \mathbf{i}\sin\frac{2\pi}{m}$ denotes a principal m-th root of 1, $\bar{x}$ denotes the complex conjugate of $x \in \mathbf{C}$. The vector $\mathbf{b} = (b_j) \in \mathbf{C}^m$ such that $b_j = \sum_{i=0}^{m-1} a_i \omega_m^{ij}$ is called the discrete Fourier transform of the vector $\mathbf{a} = (a_j)$, and is denoted by $\mathbf{b} = \mathrm{DFT}_m(\mathbf{a})$, the vector $\mathbf{a} = \mathrm{IDFT}_m(\mathbf{b})$, $\mathbf{a} = (a_j)$, $a_j = \frac{1}{m}\sum_{i=0}^{m-1} \bar{\omega}_m^{ij} b_i$ denotes the inverse discrete Fourier transform of $\mathbf{b}$.

We recall two known equations relating the vectors $\mathbf{w}, \mathbf{z} \in \mathbf{C}^m$ such that $\mathbf{z} = \mathrm{DFT}_m(\mathbf{w})$, on which the algorithms of Cooley-Tukey and Sande-Tukey are based. Let

$$\begin{array}{l} D = \mathrm{diag}(1, \omega_m, \omega_m^2, \ldots, \omega_m^{m/2-1}), \\ \mathbf{w}^{(1)} = (w_j),\ \mathbf{w}^{(2)} = (w_{m/2+j}),\ \mathbf{z}^{(even)} = (z_{2j}),\ \mathbf{z}^{(odd)} = (z_{2j+1}) \in \mathbf{C}^{m/2} \end{array} \tag{14}$$

then

$$\begin{cases} \mathbf{z}^{(even)} = \mathrm{DFT}_{m/2}(\mathbf{w}^{(1)} + \mathbf{w}^{(2)}) \\ \mathbf{z}^{(odd)} = \mathrm{DFT}_{m/2}(D(\mathbf{w}^{(1)} - \mathbf{w}^{(2)})) \end{cases} \tag{15}$$

and

$$\begin{cases} \mathbf{w}^{(1)} = (\ \mathrm{IDFT}_{m/2}(\mathbf{z}^{(even)}) + \bar{D}\ \mathrm{IDFT}_{m/2}(\mathbf{z}^{(odd)}))/2 \\ \mathbf{w}^{(2)} = (\ \mathrm{IDFT}_{m/2}(\mathbf{z}^{(even)}) - \bar{D}\ \mathrm{IDFT}_{m/2}(\mathbf{z}^{(odd)}))/2 \end{cases} \tag{16}$$

where $\bar{D}$ is the complex conjugate of D.

The following problem, which consists in computing the DFT of a real vector of m components, is solved by means of the computation of a complex DFT of order $m/2$.

Problem 1. *Given the integer $m = 2^q$, $q > 0$ integer, and the real coefficients $a_0, a_1, \ldots, a_{m-1}$ of the polynomial $a(x) = \sum_{j=0}^{m-1} a_j x^j$, compute the values $b_j = a(\omega_m^j)$, $j = 0, \ldots, m-1$ that the polynomial $a(x)$ takes on at the m-th roots of 1.*

Solution. The following well-known formulae reduce Problem 1 to computing a complex DFT of order $m/2$. Let

$$\begin{array}{l} \mathbf{w} = (w_j), w_j = a_{2j} + \mathbf{i}a_{2j+1},\ j = 0, \ldots, m/2-1, \\ \mathbf{z} = \mathrm{DFT}_{m/2}(\mathbf{w}), \end{array} \tag{17}$$

then

$$
\begin{array}{l}
b_0 = \mathrm{re}(z_0) + \mathrm{im}(z_0),\ b_{m/2} = \mathrm{re}(z_0) - \mathrm{im}(z_0), \\
b_j = ((z_j + \bar{z}_{m/2-j}) - \mathbf{i}\omega_m^j (z_j - \bar{z}_{m/2-j}))/2, \\
b_{m/2+j} = \bar{b}_{m/2-j},\ j = 1, \ldots, m/2-1.
\end{array} \tag{18}
$$

□

The converse of Problem 1 is described below.

Problem 2. *Given the integer $m = 2^q$ as in Problem 1, and the values $b_j = a(\omega_m^j)$, $j = 0, \ldots, m-1$ that the real polynomial $a(x) = \sum_{j=0}^{m-1} a_j x^j$ takes on at the m-th roots of 1, compute the coefficients $a_0, a_1, \ldots, a_{m-1}$ of $a(x)$.*

Solution. Problem 2 can be easily reduced to the computation of a complex IDFT of order $m/2$; in fact, we have the following formulae that can be obtained by reverting (17) and (18). Let

$$
\begin{array}{l}
\mathbf{z} = (z_j), \\
z_j = ((b_j + b_{m/2+j}) + \mathbf{i}\bar{\omega}_m^j (b_j - b_{m/2+j}))/2,\ j = 0, \ldots, m/2-1, \\
\mathbf{w} = \mathrm{IDFT}_{m/2}(\mathbf{z}),
\end{array} \tag{19}
$$

then

$$
a_{2j} = \mathrm{re}(w_j),\ a_{2j+1} = \mathrm{im}(w_j),\ j = 0, \ldots, m/2-1. \tag{20}
$$

□

The next problem consists in computing the values that a polynomial of degree less than m takes on at the odd powers of ω_{2m}, once the values taken on at the even powers of ω_{2m} have been computed.

Problem 3. *Given the integer $m = 2^q$, as in Problem 1, the coefficients $a_0, a_1, \ldots, a_{m-1}$ of the polynomial $a(x)$ and the values $b_{2j} = a(\omega_{2m}^{2j}) = a(\omega_m^j)$, $j = 0, \ldots, m-1$ that $a(x)$ takes on at the even powers of ω_{2m}, compute the values $b_{2j+1} = a(\omega_{2m}^{2j+1})$, $j = 0, \ldots, m-1$, that $a(x)$ takes on at the odd powers of ω_{2m}.*

Solution. Indeed, Problem 3 can be solved by computing the DFT of order $2m$ of the real vector $\mathbf{u} = (\mathbf{u}^{(1)T}, \mathbf{u}^{(2)T})^T$, where $\mathbf{u}^{(1)} = (a_0, \ldots, a_{m-1})^T$, $\mathbf{u}^{(2)} = \mathbf{0}$, i.e., $\mathbf{v} = \mathrm{DFT}_{2m}(\mathbf{u})$. But in this way the values b_{2j} would be computed once again. In a more efficient way this problem can be solved by reducing it to the computation of a complex DFT of order $m/2$. In fact, by using (17) and (18) where m is replaced by $2m$, we find that Problem 3 is reduced to computing $\mathbf{z} = \mathrm{DFT}_m(\mathbf{w})$ where the vector $\mathbf{w} = (w_j)$ is such that $w_j = a_{2j} + \mathbf{i}a_{2j+1}$, $w_{m/2+j} = 0$, $j = 0, \ldots, m/2-1$. Partitionig $\mathbf{w}$ as $(\mathbf{w} = (\mathbf{w}^{(1)T}, \mathbf{w}^{(2)T})^T$ and applying (14) and (15) yields

$$
\begin{array}{l}
\mathbf{z}^{(even)} = \mathrm{DFT}_{m/2}(\mathbf{w}^{(1)}) \\
\mathbf{z}^{(odd)} = \mathrm{DFT}_{m/2}(D\mathbf{w}^{(1)})
\end{array}
$$

since $\mathbf{w}^{(2)} = \mathbf{0}$. Now, again from (18) we find that the components b_{2j+1}, are fully defined by the components z_{2j+1}. In this way the computation is reduced to a single DFT of order $m/2$, namely $\mathrm{DFT}_{m/2}(D\mathbf{w}^{(1)})$. □

In the computation of the coefficients of the power series $\varphi^{(n+1)}(z)$, $\widehat{\varphi}^{(n+1)}(z)$ (7) by means of the evaluation/interpolation technique, we also need to solve the following problem.

Problem 4. *Given the integer $m = 2^q$ as in Problem 1, the values $b_{2j+1} = a(\omega_{2m}^{2j+1})$, $j = 0, \ldots, m-1$ that the real polynomial $a(x) = \sum_{j=0}^{2m-1} a_j x^j$ takes on at the odd powers of ω_{2m}, and $\mathbf{a}^{(1)} = \mathrm{IDFT}_m(\mathbf{b}^{(even)})$, $\mathbf{b}^{(even)} = (b_{2j})$, $b_{2j} = a(\omega_{2m}^{2j}) = a(\omega_m^j)$, $j = 0, \ldots, m-1$, compute the coefficients $a_0, a_1, \ldots, a_{2m-1}$ of $a(x)$.*

Solution. Indeed, this problem can be solved by computing $\mathbf{a} = \mathrm{IDFT}_{2m}\mathbf{b}$. A cheaper solution can be obtained as follows. By applying (19) and (20) where m is replaced by $2m$, we find that the vector $\mathbf{a}$ is recovered from the components of $\mathbf{w}$, where $\mathbf{w} = \mathrm{IDFT}_m(\mathbf{z})$. Now, in order to compute $\mathbf{w}$ we apply (14), (16) and reduce the problem to computing $\mathrm{IDFT}_{m/2}(\mathbf{z}^{(even)})$ and $\mathrm{IDFT}_{m/2}(\mathbf{z}^{(odd)})$. Now, the vector $\mathrm{IDFT}_{m/2}(\mathbf{z}^{(even)})$ is already available and needs the computation of no IDFT. In fact, observe that applying (19) and (20) for the computation of $\mathbf{a}^{(1)} = \mathrm{IDFT}_{m/2}(\mathbf{b}^{(even)})$, i.e., with a_j and b_j replaced with $a_j^{(1)}$ and b_{2j}, respectively, yields $\mathrm{IDFT}_{m/2}(\mathbf{z}^{(even)})$ as explicit function of $a_j^{(1)}$, more precisely, $\mathrm{IDFT}(\mathbf{z}^{(even)}) = (a_{2j}^{(1)} + \mathbf{i}a_{2j+1}^{(1)})$. □

5 Implementation of the algorithm and numerical results

By using the results presented in Sections 3 and 4 we may now give a more accurate description of Algorithm 2.1 in the form of the following:

Algorithm 5.1. *Computation of the solution G of (3).*

Input. Positive integers q_0, M_0, k, $M_0 = 2^{q_0}$, an error bound $\epsilon > 0$, and the nonnegative $k \times k$ matrices A_i, $i = 0, 1, \ldots, M_0$, such that the matrix $\sum_{i=0}^{M_0} A_i$ is ϵ-stochastic and the associated Markov chain is positive recurrent.

Output. An approximation $\widetilde{G}$ of the solution G of (3) together with an error bound δ such that $\mathbf{e}^T|\sum_{i=0}^{+\infty} \widetilde{G}^i A_i - \widetilde{G}| \le \delta \mathbf{e}^T$.

Computation. 1. *Initialization.*
Let $\varphi^{(0)}(z) = \sum_{i=0}^{M_0} A_i z^i$, $\widehat{\varphi}^{(0)}(z) = \sum_{i=0}^{M_0-1} A_{i+1} z^i$, $R^{(0)} = A_0(I - \sum_{i=1}^{M_0} A_i)^{-1}$, $n = 0$.

2. *Computation of the coefficients of the matrix polynomials* $P(z)$ *and* $\widehat{P}(z)$ *of degree* M_n *and* $M_n - 1$ *which approximate the matrix power series* $\varphi^{(n)}(z)$ *and* $\widehat{\varphi}^{(n)}(z)$.

Repeat

(a) Set $n = n + 1$, $q_n = q_{n-1} - 1, M_n = 2^{q_n}$.

(b) Compute the mean values $\boldsymbol{\alpha}^{(n)}$ and $\widehat{\boldsymbol{\alpha}}^{(n)}$ by means of (11).

(c) Evaluate the functions $\varphi_{odd}^{(n-1)}(z)$, $\varphi_{even}^{(n-1)}(z)$ and $\widehat{\varphi}_{odd}^{(n-1)}(z)$, $\widehat{\varphi}_{even}^{(n-1)}(z)$ at the set of Fourier points

$$\mathcal{F}_{M_n} = \{\omega_{M_n}^i, \quad i = 0, 1, \ldots, M_n - 1\}.$$

This computation is performed by means of the solution of Problem 1, if $M_n = 2^{q_{n-1}-1}$, that is, when this evaluation stage is encountered for the first time. The computation is performed by means of the solution of Problem 3, otherwise.

(d) Pointwise apply equations (7), that is, successively compute the matrices

$$\begin{aligned} S_1(z) &= \varphi_{even}^{(n-1)}(z)(I - \varphi_{odd}^{(n-1)}(z))^{-1} \\ S_2(z) &= z\varphi_{odd}^{(n-1)}(z) + S_1(z)\varphi_{even}^{(n-1)}(z) \\ \widehat{S}_2(z) &= \widehat{\varphi}_{odd}^{(n-1)}(z) + S_1(z)\widehat{\varphi}_{even}^{(n-1)}(z) \end{aligned} \tag{21}$$

for $z \in \mathcal{F}_{M_n}$ if $M_n = 2^{q_{n-1}-1}$, that is, when this stage is encountered for the first time; otherwise for $z \in \mathcal{F}_{M_n} - \mathcal{F}_{M_n/2}$.

(e) Interpolate the values of $S_2(z)$ and $\widehat{S}_2(z)$ at $z \in \mathcal{F}_{M_n}$, and obtain the coefficients of the matrix polynomials $P(z)$ and $\widehat{P}(z)$. This computation is performed by means of the solution of Problem 2, if $M_n = 2^{q_{n-1}-1}$, that is, when this interpolation stage is encountered for the first time. The computation is performed by means of the solution of Problem 4, otherwise.

(f) Apply tests (12) and (13) in order to check if $P(z)$ and $\widehat{P}(z)$ are good approximations of the series. If the inequalities (12) and (13) are satisfied then skip to the next stage. Otherwise, set $M_n = 2M_n$, $q_n = q_n + 1$ and repeat from stage (2c).

(g) Set $\varphi^{(n)}(z) = P(z)$, $\widehat{\varphi}^{(n)}(z) = \widehat{P}(z)$.

(h) Compute $R^{(n)} = A_0^{(n)}(I - \sum_{i=1}^{M_n} A_i^{(n)})^{-1}$.

Until one of the following conditions is verified.

$$|R^{(n)} - R^{(n-1)}| < \epsilon E, \tag{C1}$$
$$\mathbf{e}^T(I - A_0^{(n)}(I - A_1^{(n)})^{-1}) < \epsilon \mathbf{e}^T, \tag{C2}$$
$$\mathbf{e}^T(I - A_0(I - \widehat{A}_1^{(n)})^{-1}) < \epsilon \mathbf{e}^T. \tag{C3}$$

3. *Computation of the matrix* $\widetilde{G}$.
   (a) If condition (C1) is verified then $\widetilde{G} = A_0(I - \sum_{i=0}^{\widehat{M}_n - 1} {R^{(n)}}^i \widehat{A}_{i+1}^{(n)})^{-1}$.
   (b) If condition (C2) is verified then $\widetilde{G} = A_0(I - \widehat{A}_1^{(n)})^{-1}$.
   (c) If condition (C3) is verified then $\widetilde{G} = A_0(I - \widehat{A}_1^{(n)})^{-1}$.
4. *Computation of the error bound* δ.
   Compute $\delta = \max_{n=1,\ldots,k} \sum_{i=1}^{k} |w_{i,j}|$, for $W = (w_{i,j})_{i,j}$, $W = \sum_{i=0}^{M_0} \widetilde{G}^i A_i - \widetilde{G}$.

The above algorithm has been implemented in Fortran90 in the subroutine `pwcr` (pointwise cyclic reduction). In order to use the subroutine `pwcr`, a driver program `pwcr_drv` is also provided. This program `pwcr_drv` reads from the standard input the file name `<input_file_name>` of the input data. This file must contain the folling data on each line:

1. On the first line: the block dimension k;
2. On the second line: the number $M_0 + 1$ of blocks A_i such that $\sum_{i=0}^{M_0} A_i$ is numerically stochastic;
3. On the third line: the error bound ϵ;
4. On the subsequent lines: the nonzero entries $(A_i)_{r,s}$ of the blocks A_i coded as follows: $i+1$, r, s, $(A_i)_{r,s}$;
5. On the last line: the quartuple $0, 0, 0, 0.d0$, which denotes the end of the data file.

The driver calls the subroutine `pwcr` and writes the output into the file `<input_file_name>.out`. The output file contains the residual error $|| \sum_{i=0}^{M_0} \widetilde{G}^i A_i - \widetilde{G}||_1$, where $\widetilde{G}$ is the computed approximation of the solution G of (3), and the entries $\widetilde{g}_{r,s}$ of $\widetilde{G}$ arranged row-wise.

The subroutine `pwcr` has the call sequence

```
CALL pwcr(a,eps,g,err)
```

where: `a` is a real(kind(0.d0)), dimension(:,:,:), pointer, containing the entries of the blocks A_i; `eps` is a real(kind(0.d0)) containing the error bound ϵ; `g` is a real(kind(0.d0)), dimension(:,:), pointer, containing the entries of the computed approximation $\widetilde{G}$; `err` is the residual error $|| \sum_{i=0}^{M_0} \widetilde{G}^i A_i - \widetilde{G}||_1$.

The main subroutine used inside `pwcr` is the subroutine `schur` which performs one step of cyclic reduction by computing the even and the odd components of the matrix series $\varphi^{(n+1)}(z)$ and $\widehat{\varphi}^{(n+1)}(z)$ (compare (7)) which define the Schur complement in the cyclic reduction process.

In order to compute the values of the series $\varphi^{(n+1)}(z)$ and $\widehat{\varphi}^{(n+1)}(z)$ at the roots of 1, several subroutines which implement the solutions of problems 1,2,3,4, of section 4 in the case of matrix polynomials, have been introduced. More

specifically the subroutines `ftb1` and `iftb1` solve problems 1 and 2, respectively; the subroutines `ftb2` and `iftb2` solve problems 3 and 4, respectively.

For the computation of the FFT's of real vectors, the split radix algorithm of Dhuamel-Hollman, implemented by Sorensen et al [19], has been modified in order to avoid the recomputation of the Fourier points. The subroutines that are involved in this computation are: `fft1`, `ifft1`, `ffts1`, `iffts1`, `ffts2`, `iffts2`, `twiddle`, `itwiddle`. The Fourier points are computed by the subroutine `fillroots`.

The subroutines involved in the computation of the FFT's are contained in the file `pwcr_fft.f90`. The subroutines `pwcr, schur`, and the related subroutines, are contained in the file `pwcr_sub.f90`.

We have tested our algorithm on a problem arising from the mathematical modeling of a Metaring MAC Protocol [1]. For this particular example the blocks A_i have dimension $k = 16$. We have tested the algorithm for different values of the parameter ρ, $0 < \rho < 1$, which represents the stability condition of the associated Markov chain [16]: as ρ tends to one, the problem of computing the solution G of (3) by using customary techniques becomes more difficult; in fact, for $\rho = 1$ the Markov chain is not positive recurrent [16]. The numerical degree of the series $\varphi^{(0)}(z)$ is equal to 168 for $\rho = 0.1$, is equal to 240 for $\rho = 0.8$, and is equal to 264 for the remaining tested values of $\rho \geq 0.9$. We have compared our algorithm with the cyclic reduction of [3] and with the functional iteration method based on the recursion

$$\begin{aligned} &X_0 = I, \\ &X_{n+1} = A_0 \left(I - \sum_{i=2}^{+\infty} X_n^{i-1} A_i \right)^{-1}, \ n \geq 0, \end{aligned} \tag{22}$$

which is the fastest among the classical linearly convergent functional iterations [15].

The numerical experiments have been performed on an alpha workstation with a base 2 arithmetic endowed with 53 bits.

Table 1 reports the CPU time (in seconds) and the number of iterations needed by the Point-Wise Cyclic Reduction (PWCR) algorithm to compute an approximation of the matrix G, by choosing $\epsilon = 10^{-12}$; the residual of the computed approximation is also reported.

Table 2 reports the CPU time (in seconds) and the number of iterations needed by the Cyclic Reduction (CR) implemented in [3], the residual of the computed approximation, and the ratio between the times needed by CR and by PWCR. Observe that the algorithm PWCR, besides being faster than CR, leads to a higher accuracy of the result.

We also compared our method with classical techiques based on functional iterations: Table 3 reports the CPU time (in seconds) and the number of iterations needed by the Functional Iteration Formula (FIF) defined in (22), the residual of the computed approximation, and the ratio between the times needed by this algorithm and by PWCR.

ρ	Time (s.)	Iterations	Residual
0.1	0.9	9	$1.8 \cdot 10^{-13}$
0.8	1.5	13	$5.4 \cdot 10^{-14}$
0.9	2.3	14	$1.5 \cdot 10^{-14}$
0.95	2.4	16	$1.8 \cdot 10^{-14}$
0.96	2.4	17	$2.3 \cdot 10^{-14}$
0.97	2.5	20	$4.2 \cdot 10^{-14}$

Table 1: Point-Wise Cyclic Reduction

ρ	Time (s.)	Iterations	Residual	CCR/PWCR
0.1	1.3	9	$8.4 \cdot 10^{-14}$	1.4
0.8	3.2	13	$2.7 \cdot 10^{-13}$	2.1
0.9	3.3	14	$2.7 \cdot 10^{-13}$	1.4
0.95	3.4	16	$1.8 \cdot 10^{-13}$	1.4
0.96	3.4	17	$2.0 \cdot 10^{-13}$	1.4
0.97	3.5	20	$2.3 \cdot 10^{-13}$	1.4

Table 2: Customary Cyclic Reduction

Table 3 shows the effectiveness of our method, specially in cases where customary techniques converge very slowly. Indeed, observe that the cyclic reduction algorithm is almost insensitive to the values of the parameter ρ, whereas the performance of the functional iteration method strongly deteriorates as ρ approaches to 1.

References

[1] G. Anastasi, L. Lenzini, B. Meini, "Performance evaluation of a worst case model of the Metaring MAC Protocol with global fairness", *Performance Evaluation,* to appear.

[2] D. Bini, B. Meini, "On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems", *Computations with Markov Chains*, W. J. Stewart Ed., pp. 21–38, Kluwer Academic Publishers, 1996.

[3] D. Bini, B. Meini, "On the solution of a nonlinear matrix equation arising in queueing problems", *SIAM J. Matrix Anal. Appl.,* 17 (1996), pp. 906–926.

ρ	Time (s.)	Iterations	Residual	FIF/PWCR
0.1	0.3	22	$2.2 \cdot 10^{-14}$	0.3
0.8	3.9	148	$1.1 \cdot 10^{-13}$	2.6
0.9	10.8	373	$1.2 \cdot 10^{-13}$	4.7
0.95	44.4	1534	$1.3 \cdot 10^{-13}$	18.5
0.96	91.4	3158	$1.3 \cdot 10^{-13}$	38.1
0.97	96.8	3336	$1.3 \cdot 10^{-13}$	38.7

Table 3: Functional Iteration Method

[4] D. Bini, V. Pan, *Matrix and Polynomial Computations, Vol. 1: Fundamental Algorithms,* Birkhäuser, Boston 1994.

[5] D. Bini, V. Pan, "Polynomial division and its computational complexity", *J. Complexity,* 6 (1986), pp. 179–203.

[6] E. O. Brighman, *The Fast Fourier Transform*, Prentice Hall, Englewood Cliffs, N. J., 1974.

[7] B. L. Buzbee, G. H. Golub, C. W. Nielson, "On direct methods for solving Poisson's equation", *SIAM J. Numer. Anal.,* 7 (1970), pp. 627–656.

[8] E. Çinlar, *Introduction to Stochastic Processes,* Prentice Hall, Englewood Cliffs, N.J., 1975.

[9] D.F. Elliott, K.R. Rao, *Fast Transform Algorithms, Analyses, Applications,* Academic Press, New York, 1982.

[10] G. Latouche, "Algorithms for evaluating the matrix G in Markov chains of PH/G/1 type", Bellcore Technical Report, 1992.

[11] G. Latouche, "Newton's iteration for non-linear equations in Markov chains", *IMA J. of Numerical Analysis,* 14 (1994), pp. 583–598.

[12] G. Latouche, V. Ramaswami, "A logarithmic reduction algorithm for quasi-birth-death processes", *J. Appl. Prob.,* 30 (1993), pp. 650–674.

[13] G. Latouche, G. W. Stewart, "Numerical methods for M/G/1 type queues", *Computations with Markov Chains*, W. J. Stewart Ed., pp. 571–581, Kluwer Academic Publishers, 1996.

[14] B. Meini, "An improved FFT-based version of the Ramaswami formula", *Comm. Stat. Stochastic Models, 13 (1997),* to appear.

[15] B. Meini, "New convergence results on functional iteration techniques for the numerical solution of M/G/1 type Markov chains", *Numer. Math.,* to appear.

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

[17] V. Ramaswami, "A stable recursion for the steady state vector in Markov chains of $M/G/1$ type", *Commun. Statist. Stochastic Models,* 4 (1988), pp. 183-188.

[18] V. Ramaswami, "Nonlinear matrix equations in applied probability - Solution techniques and open problems", *SIAM Review,* 30 (1988), pp. 256-263.

[19] M. Sorensen, M. Heideman, C. Burrus, "On computing the split-radix FFT", *IEEE Trans. Acoust. Signal Processing,* 34 (1986), pp. 152–156.

[20] G. W. Stewart, "On the solution of block Hessenberg systems", *Numer. Lin. Algebra with Appl.*, 2 (1995), pp. 287–296.

On functional iteration methods for solving nonlinear matrix equations arising in queueing problems

Paola Favati* Beatrice Meini†

Abstract

The problem of the computation of the minimal nonnegative solution G of the nonlinear matrix equation $X = \sum_{i=0}^{+\infty} X^i A_i$ is considered. This problem arises in the numerical solution of M/G/1 type Markov chains, where A_i, $i \geq 0$, are nonnegative $k \times k$ matrices such that $\sum_{i=0}^{+\infty} A_i$ is column stochastic. We analyze classical functional iteration methods, by estimating the rate of convergence, in relation with spectral properties of the starting approximation matrix X_0. Based on these new convergence results, we propose an effective method to choose a matrix X_0, which drastically reduces the number of iterations; the additional cost needed to compute X_0 is much less than the overall savings achieved by reducing the number of iterations.

1 Introduction

Many queueing problems can be modeled by Markov chains of M/G/1 type, that is, Markov chains whose probability transition matrix has the structure

$$P^t = \begin{pmatrix} B_1 & A_0 & & \bigcirc & \\ B_2 & A_1 & A_0 & & \\ B_3 & A_2 & A_1 & A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix}, \tag{1}$$

where B_{i+1}, A_i, $i \geq 0$, are $k \times k$ nonnegative matrices such that $\sum_{i=1}^{+\infty} B_i$ and $\sum_{i=0}^{+\infty} A_i$ are column stochastic. One of the major problems in Markov chains is the computation of the probability invariant vector $\boldsymbol{\pi}$, i.e., the solution of

*Istituto di Matematica Computazionale del C.N.R., via S. Maria 46, 56127 Pisa, Italy. E-mail: favati@imc.pi.cnr.it

†Dipartimento di Matematica, via Buonarroti 2, 56127 Pisa, Italy. E-mail: meini@dm.unipi.it

the infinite system $\boldsymbol{\pi} = P^t\boldsymbol{\pi}$, $\|\boldsymbol{\pi}\|_1 = 1$. For the matrices of structure (1), the computation of $\boldsymbol{\pi}$ can be reduced (compare [16]) to the computation of the minimal nonnegative solution G of the nonlinear matrix equation

$$X = \sum_{i=0}^{+\infty} X^i A_i, \tag{2}$$

where X is a $k \times k$ matrix. Indeed, once G is known, the vector $\boldsymbol{\pi}$ can be recovered by means of the Ramaswami formula [17]. Moreover, the knowledge of the matrix G is very important, due to its role in calculating performance measures [16].

Recently, the analysis of numerical methods for computing the solution of the matrix equation (2) has been largely developed. New quadratically convergent methods have been proposed in [12], [13], [11], [3], [4], [5]; classical linearly convergent methods based on functional iterations have been improved and studied in details in [8], [14]. The latter methods, unlike most of the quadratically convergent methods, are very easy to implement. They are based on the recursion

$$X_{n+1} = F(X_n), \quad n \geq 0, \tag{3}$$

where

$$F(X) = K(X)(I - H(X))^{-1}, \tag{4}$$

and $H(X) = \sum_{i=0}^{+\infty} X^i H_i$, $K(X) = \sum_{i=0}^{+\infty} X^i K_i$ are such that $H_i \geq O$, $K_i \geq O$, $i \geq 0$, and $\sum_{i=0}^{+\infty} z^i A_i = z\sum_{i=0}^{+\infty} z^i H_i + \sum_{i=0}^{+\infty} z^i K_i$ [8], [14]. It can be proved that, by suitable choosing the starting approximation X_0, the sequence (3) linearly converges to the solution G of (2); moreover, the rate of convergence is strongly related to the choice of X_0. More precisely, we associate with any function $F(\cdot)$ of (4) a suitable nonnegative $k^2 \times k^2$ matrix $\widehat{R}_F$. In the case where $X_0 = 0$ in [14] it is shown that the asymptotic rate of convergence of $\{X_n\}$ coincides with the spectral radius of $\widehat{R}_F$. Moreover, in the case where X_0 is any column stochastic matrix, it is shown that at each step n, the vector $\hat{\mathbf{e}}_n$, obtained by the column-wise arrangement of the entries of the matrix $G - X_n$, is orthogonal to the nonnegative left eigenvector $\mathbf{v}^T$ associated with the spectral radius of the matrix $\widehat{R}_F$. From this property, it follows that the rate of convergence of X_n is less than or equal to the spectral radius of the matrix M_1 obtained by projecting $\widehat{R}_F$ onto a suitable subspace $\mathcal{V}_1$.

From the above results it follows that, if the matrix $\widehat{R}_F$ is irreducible, then the mean asymptotic rate of convergence of the sequence X_n is:

– equal to the spectral radius of $\widehat{R}_F$, if $X_0 = O$;

– less than or equal to the second largest modulus eigenvalue of $\widehat{R}_F$, if X_0 is a column stochastic matrix.

The better rate of convergence, when X_0 is a stochastic matrix, is mainly due to the fact that both G and X_0 have the eigenvalue 1 and associated left eigenvector $\mathbf{e}^t = (1, \ldots, 1)$.

In this paper, in order to further improve the rate of convergence, we consider starting approximations X_0 that share with G more eigenvalues, and the corresponding left eigenvectors, in addition to the eigenvalue 1 and the eigenvector $\mathbf{e}^t$.

More precisely, let $\lambda_1 = 1, \lambda_2, \ldots, \lambda_h$, $h \leq k$, be eigenvalues of the matrix G such that the corresponding left eigenvectors $\mathbf{w}_1^t = \mathbf{e}^t, \mathbf{w}_2^t, \ldots, \mathbf{w}_h^t$ are linearly independent. Suppose that the matrix X_0 satisfies

$$\mathbf{w}_i^t X_0 = \lambda_i \mathbf{w}_i^t, \ i = 1, \ldots, h, \tag{5}$$

that is, it shares with G the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_h$ and the corresponding left eigenvectors. Then we prove that the rate of convergence of the sequence X_n is bounded from above by $\rho(M_h)$, where $\rho(A)$ denotes the spectral radius of the matrix A and M_h is the matrix obtained by projecting $\widehat{R}_F$ onto a suitable subspace $\mathcal{V}_h$. Moreover, we show that $\rho(M_h) \leq \rho(M_1)$, for any $h \in \{1, \ldots, k\}$. In this way the rate of convergence can be substantially reduced.

Once the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_h$, and the corresponding eigenvectors $\mathbf{w}_1^t, \ldots, \mathbf{w}_h^t$ are known, a matrix X_0 satisfying (5) can simply be obtained by means of the equation $X_0 = V Diag(\lambda_1, \ldots, \lambda_h) W^t$, where W is the $k \times h$ matrix whose columns are the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_h$, and V is the generalized inverse of W^t. When $h = k$, i.e., all the eigenvalues, and the corresponding left eigenvectors, are known, the matrix V coincides with the inverse of W^t. In this case some authors [9], [15] have proposed to approximate G by simply computing $W^{-t} Diag(\lambda_1, \ldots, \lambda_k) W^t$. In practice, due to the roundoff errors generated in the computation, the matrix $\widetilde{G}$ obtained in this way can be a poor approximation of G; we propose to improve this approximation by choosing $X_0 = \widetilde{G}$ and by applying few steps of the functional iteration method (3).

We have performed several numerical experiments. We have first computed eigenvalues and corresponding left eigenvectors, for different values of h. Then a matrix X_0 satisfying (5) has been computed and functional iteration has been applied. From the many results, we have observed a drastic reduction of the number of iterations needed to approximate G when $h > 1$, with respect to the case where $h = 1$, or X_0 is a generic column stochastic matrix. On the other hand, the time needed to approximate the eigenvalues, and the corresponding left eigenvectors, when they are not known, is negligible with respect to the time needed to perform functional iterations.

The paper is organized as follows: in Section 2 we prove the main convergence results, by estimating the rate of convergence, in relation with the starting approximation X_0; in Section 3 we propose a method for computing a matrix X_0 satisfying (5); in Section 4 we present numerical results and comparisons.

2 Convergence rate and starting approximation

In this section we extend the convergence results proved in [14] for functional iteration methods based on the recursion (3), for solving the nonlinear matrix equation (2); these new convergence results suggest how to choose the starting approximation matrix X_0 in order to drastically reduce the number of iterations needed to approximate the solution G.

Let us denote by $\hat{\mathbf{e}}_n$ the vector obtained by column-wise arranging the entries of the matrix $G - X_n$. It can be shown [14] that the error $\hat{\mathbf{e}}_{n+1}$ is related to the error $\hat{\mathbf{e}}_n$ by means of the recursion

$$\hat{\mathbf{e}}_{n+1} = \widehat{R}_n \hat{\mathbf{e}}_n \tag{6}$$

where

$$\widehat{R}_n = \sum_{j=0}^{+\infty} Y_{j,n} \otimes G^j \tag{7}$$

and

$$\begin{aligned} Y_{0,n} &= \textstyle\sum_{i=1}^{+\infty} \left(X_n^{i-1} K_i (I - H(X_n))^{-1} \right)^t \\ Y_{j,n} &= \textstyle\sum_{i=j}^{+\infty} \left(X_n^{i-j} A_{i+1} (I - H(X_n))^{-1} \right)^t, \quad j \geq 1. \end{aligned}$$

Denote by

$$\widehat{R}_F = \sum_{j=0}^{+\infty} Y_j \otimes G^j$$

where

$$\begin{aligned} Y_0 &= \textstyle\sum_{i=1}^{+\infty} \left(G^{i-1} K_i (I - H(G))^{-1} \right)^t \\ Y_j &= \textstyle\sum_{i=j}^{+\infty} \left(G^{i-j} A_{i+1} (I - H(G))^{-1} \right)^t, \quad j \geq 1. \end{aligned}$$

Observe that, if $\lim_n X_n = G$, then $\widehat{R}_F = \lim_n \widehat{R}_n$. In the following we denote by $\mathbf{e}_i$, $i = 1, \ldots, k$, the k-dimensional vector having the i-th entry equal to 1, and the remaining ones equal to 0.

Relation (6) allows us to prove the following convergence result [14]:

Theorem 1 *Let*

$$r = \lim_n \sqrt[n]{\|\hat{\mathbf{e}}_n\|}$$

be the mean asymptotic rate of convergence of the sequence $X_{n+1} = F(X_n)$, where $F(\cdot)$ is defined by (4) and $\|\cdot\|$ is any vector norm. The following results hold:

– if $X_0 = 0$ then $r = \rho(\widehat{R}_F)$;

– if X_0 is a stochastic matrix then $r \leq \rho(T_1^t \widehat{R}_F T_1)$, where T_1 is the $k^2 \times (k^2 - k)$ matrix whose columns complete the set of vectors

$$\frac{1}{\sqrt{k}} (\mathbf{e}_i \otimes \mathbf{e}), \quad i = 1, \ldots, k, \tag{8}$$

to an orthonormal basis of $\mathbf{R}^{k^2}$.

The reduction of the rate of convergence, in the case where X_0 is a column stochastic matrix, is mainly due to the fact that X_0 has the eigenvalue 1 and associated left eigenvector $\mathbf{e}^t$, as well as the matrix G. In fact, from this property it follows that the vector $\hat{\mathbf{e}}_n$ is orthogonal to the nonnegative left eigenvector associated with the spectral radius of the matrix $\widehat{R}_F$.

This fact suggests us to choose a starting approximation X_0 that shares with G some other eigenvalues and the corresponding left eigenvectors, in addition to the eigenvalue equal to 1 and the eigenvector $\mathbf{e}^t$.

Denote by $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ the set of the eigenvalues of the matrix G, and with $\mathbf{w}_1^t, \ldots, \mathbf{w}_k^t$ the corresponding left eigenvectors, where $\lambda_1 = 1$ and $\mathbf{w}_1^t = \mathbf{e}^t$. Let Λ_h a subset of h eigenvalues of Λ, $1 \leq h \leq k$, such that:

1. $\lambda_1 \in \Lambda_h$;
2. if $\lambda \in \Lambda_h$, then $\bar{\lambda} \in \Lambda_h$, where $\bar{\lambda}$ denotes the complex conjugate of λ;
3. the left eigenvectors corresponding to the eigenvalues in Λ_h are linearly independent.

Throughout, we suppose, without loss of generality, that $\Lambda_h = \{\lambda_1, \ldots, \lambda_h\}$.

Denote by $\mathcal{U}_h$ the subspace generated by the k^2–dimensional vectors $\mathbf{e}_i \otimes \mathbf{w}_j$, $i = 1, \ldots, k$, $j = 1, \ldots, h$, and let $\mathcal{V}_h = \mathcal{U}_h^{\perp}$. Let $S_h = I \otimes E_h$ a $k^2 \times hk$ matrix whose columns are an orthonormal basis of $\mathcal{U}_h$, and let $T_h = I \otimes F_h$ a $k^2 \times (k^2 - hk)$ matrix whose columns are an orthonormal basis of $\mathcal{V}_h$. Then the following result holds:

Theorem 2 *Let X_n, $n \geq 0$, be a convergent sequence generated by $X_{n+1} = F(X_n)$, where $F(\cdot)$ is defined by (4). Denote by*

$$r = \lim_n \sqrt[n]{\|\hat{\mathbf{e}}_n\|}$$

the mean asymptotic rate of convergence, where $\|\cdot\|$ is any vector norm. Let $1 \leq h \leq k$ and $\Lambda_h \subset \Lambda$ satisfying properties 1, 2, 3. Suppose that X_0 verifies

$$\mathbf{w}^t X_0 = \lambda \mathbf{w}^t, \tag{9}$$

for any $\lambda \in \Lambda_h$, where $\mathbf{w}^t$ is the corresponding left eigenvector of G. Then it holds $r \leq \rho(T_h^t \widehat{R}_F T_h)$.

Proof. First observe that $\hat{\mathbf{e}}_n \in \mathcal{V}_h$, for $n \geq 0$. In fact, for $n = 0$, this property follows from the relations $\mathbf{w}^t(G - X_0) = \lambda \mathbf{w}^t - \lambda \mathbf{w}^t = 0$, for any $\lambda \in \Lambda_h$; for $n \geq 1$, the property follows from (6) and from the fact that, if $\mathbf{u} \in \mathcal{U}_h$, then $\mathbf{u}^t \widehat{R}_n \in \mathcal{U}_h$, for any $n \geq 0$.

Let $P = (S_h | T_h)$. From (6) it follows that

$$P^t \hat{\mathbf{e}}_{n+1} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f}_{n+1} \end{pmatrix} = (P^t \widehat{R}_n P)(P^t \hat{\mathbf{e}}_n) = (P^t \widehat{R}_n P) \begin{pmatrix} \mathbf{0} \\ \mathbf{f}_n \end{pmatrix}, \tag{10}$$

where $\mathbf{f}_n = T_h^t \mathbf{e}_n$. Since $S_h^t \widehat{R}_n T_h = O$, it follows that

$$P^t \widehat{R}_n P = \begin{pmatrix} S_h^t \\ - \\ T_h^t \end{pmatrix} \widehat{R}_n \ (S_h | T_h) = \begin{pmatrix} V_n & O \\ T_n & Z_n \end{pmatrix}, \tag{11}$$

where $Z_n = T_h^t \widehat{R}_n T_h$. From (10) and (11), the recurrence (6) can be rewritten in the form

$$\mathbf{f}_{n+1} = Z_n \mathbf{f}_n \tag{12}$$

and

$$r = \lim_n \sqrt[n]{\|\hat{\mathbf{e}}_n\|} = \lim_n \sqrt[n]{\|\mathbf{f}_n\|}.$$

By following the same lines of the proof of Theorem 15 in [14], we easily arrive at $r \leq \rho(T_h^t \widehat{R}_F T_h)$. □

From the above theorem it follows that, by starting with a matrix X_0 sharing with G more than one eigenvalue, the rate of convergence can be reduced. In fact, the following result holds:

Theorem 3 *Under the hypotheses of Theorem 2, the eigenvalues of the matrix $T_h^t \widehat{R}_F T_h$ are given by the set*

$$\bigcup_{i=h+1}^{k} \{\mu : \mu \text{ is eigenvalue of } Q_i\}$$

where $Q_i = \sum_{j=0}^{\infty} \lambda_i^j Y_j$, $i = 1, \ldots, k$.

Proof. Observe that the matrix $Z_F = T_h^t \widehat{R}_F T_h$ can be written as

$$Z_F = (I \otimes F_h^t) \widehat{R}_F (I \otimes F_h) = \sum_{j=0}^{\infty} Y_j \otimes F_h^t G^j F_h.$$

Whence, the matrix Z_F is similar to the matrix $\sum_{j=0}^{\infty} F_h^t G^j F_h \otimes Y_j$, which is similar to $\sum_{j=0}^{\infty} U^j \otimes Y_j$, where U is the Schur canonical form of $F_h^t G F_h$. Since the eigenvalues of $F_h^t G F_h$ are $\lambda_{h+1}, \ldots, \lambda_k$, U is an upper triangular matrix whose diagonal entries are $\lambda_{h+1}, \ldots, \lambda_k$. In this way we find that Z_F is similar to a block upper triangular matrix whose diagonal blocks are the matrices Q_i, $i = h+1, \ldots, k$. □

From the above theorem it follows that:

Corollary 4 *Under the hypotheses of Theorem 2 it holds that*

$$\rho(T_h^t \widehat{R}_F T_h) \leq \rho(T_1^t \widehat{R}_F T_1)$$

for any $1 \leq h \leq k$.

In practice, we are not able to establish in advance the optimal number of eigenvalues/eigenvectors h to be shared in order to obtain the best rate of convergence. The optimal number h is strongly related to the spectral properties of G, and thus to the specific problems.

In the above theorems we have presented conditions on the starting matrix X_0 to increase the rate of convergence of the sequence $X_{n+1} = F(X_n)$, under the hypothesis that X_0 is such that the sequence X_n is convergent. In the case where the matrix X_0 is stochastic, the sequence is obviously convergent. In the more general case where X_0 is not necessarily nonnegative, the local convergence is guaranteed by the classical results on functional iteration methods [18], since $\rho(\widehat{R}_F) < 1$.

3 Choosing the starting approximation

Based on the results of Section 2, in order to find a good starting matrix X_0, we need first to know eigenvalues and left eigenvectors of the matrix G. The eigenvalues $\lambda_1, \ldots, \lambda_k$ of G are given by the zeros of the function

$$\varphi(z) = \det(zI - \sum_{j=0}^{+\infty} A_j z^j) \tag{13}$$

lying in the closed unit circle of the complex plane (compare [9]), and the left eigenvectors are the solutions of the linear systems

$$\mathbf{w}_i^t(\lambda_i I - \sum_{j=0}^{+\infty} A_j \lambda_i^j) = 0, \ i = 1, \ldots, k. \tag{14}$$

The zeros of the function $\varphi(z)$ can be computed by means of any algorithm for approximating the zeros of a polynomial, for example by means of Aberth's method [7], [1]; the left eigenvectors can be computed by solving the above homogeneous linear systems.

In the applications, it may be convenient to approximate only the eigenvalues that can be computed in a few numbers of iterations, and then to compute the corresponding matrix X_0.

Suppose we know a subset $\Lambda_h = \{\lambda_1, \ldots, \lambda_h\}$ of Λ satisfying properties 1, 2, 3 of Section 2; denote by $\mathbf{w}_1^t, \ldots, \mathbf{w}_h^t$ the corresponding left eigenvectors. Let W_h be the $k \times h$ matrix whose columns are the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_h$ and let V_h be a $k \times h$ matrix such that

$$V_h^t W_h = I_h, \tag{15}$$

where I_h is the h dimensional identity matrix. Then, the matrix

$$X_0 = V_h Diag(\lambda_1, \ldots, \lambda_h) W_h^t \tag{16}$$

verifies conditions (9). A simple way to obtain a matrix V_h satisfying (15) is to compute the generalized inverse of W_h^t, say by means of QR or SVD method [10], or by means of the relation $V_h = (W_h W_h^t)^{-1} W_h$, that holds since W_h is a full rank matrix. Since $\lambda \in \Lambda_h$ implies $\bar{\lambda} \in \Lambda_h$, the matrix X_0 obtained in this way is real. If, in addition, V_h is such that

$$V_h Diag(\lambda_1, \ldots, \lambda_h) W_h^t \geq 0, \tag{17}$$

then X_0 is stochastic, since $X_0 \geq 0$ and $\mathbf{w}_1^t X_0 = \mathbf{w}_1^t$, and $\mathbf{w}_1^t = \mathbf{e}^t$.

A matrix V_h satisfying both conditions (15) and (17) can be viewed as a solution of a linear programming problem. However the computation of a solution, if it exists, can require a long time. On the other hand, in many numerical experiments, even if the starting matrix X_0, obtained by means of the generalized inverse, has negative entries after few steps the matrix X_n is nonnegative. Whence, the use of generalized inverse, leads to an advantageous choice of X_0.

Suppose we know all the eigenvalues $\lambda_1, \ldots, \lambda_k$ of G and the corresponding left eigenvectors $\mathbf{w}_1^t, \ldots, \mathbf{w}_k^t$. If the above eigenvectors are linearly independent, then it is well known [9], [15] that an approximation of the matrix G can be obtained by means of the relation

$$G = W_k^{-t} Diag(\lambda_1, \ldots, \lambda_k) W_k^t. \tag{18}$$

However, due to errors generated in the computation, the matrix obtained by calculating $W_k^{-t} Diag(\lambda_1, \ldots, \lambda_k) W_k^t$ can be a not sufficiently accurate approximation of G. In this case, in order to improve the approximation of G, we apply the functional iteration method, starting with $X_0 = W_k^{-t} Diag(\lambda_1, \ldots, \lambda_k) W_k^t$. In most applications, the matrix X_0 is nonnegative and generates a sequence X_n which rapidly converges to G.

4 Numerical results

We have performed numerical experiments in order to test the behavior of the rate of convergence with different starting approximations X_0. Since the matrix X_0 of (16) is not necessarily nonnegative, the matrices X_n could have negative entries and the matrices $I - H(X_n)$ of (4) could be singular, for small values of n. Thus, in order to avoid possibly singular matrix inversions, we use the standard functional iteration method $X_{n+1} = \sum_{i=0}^{+\infty} X_n^i A_i$, $n \geq 0$.

We have chosen matrices X_0 sharing with G one, two, or more eigenvalues and corresponding left eigenvectors. We have stopped iterations when the residual error $||X_n - \sum_{i=0}^{+\infty} X_n^i A_i||_1$ was less than $\epsilon = 10^{-11}$.

We have implemented our method in Fortran 77; the program has been run on a Ultra Sparc Workstation.

We have considered two problems; Problem 1 arises from the mathematical modeling of a metropolitan queueing network (Metaring MAC Protocol) [2],

h	Iterations	Time (s.)
1	2763	197.3
2	34	4.9
4	26	4.4
5	25	4.4
6	20	4.0
8	20	4.0
9	20	4.0
10	20	4.1
12	20	4.1
14	20	4.1
15	19	4.0
16	19	4.0

Table 1: Metaring MAC Protocol

where the blocks A_i have dimension 16; Problems 2 models a teletraffic system as a continuous time QBD Markov chain, where the blocks A_i have dimension 24.

For Problem 1 we have approximated the eigenvalues of the matrix G by means of Aberth's method, applied to the function (13), and the left eigenvectors by solving the linear systems (14). The time needed to approximate all the eigenvalues, and the corresponding left eigenvectors, was 2.5 seconds, and 0.02 seconds, respectively. We have sorted the eigenvalues in such a way that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_k|$, and we have chosen different subsets $\Lambda_h = \{\lambda_1, \ldots, \lambda_h\}$, where h is such that properties 1, 2, and 3 of Section 2 are satisfied. Table 1 reports: the number h of eigenvalues shared by X_0 with G; the number of iterations needed by the functional iteration method to reach the required accuracy; the total time (in seconds) needed to compute the matrix X_0 and to perform the functional iterations. Figure 1 reports the logarithm (to the base 10) of the residual error for $h = 1, 2, 16$.

For this problem, the time needed to reach the required accuracy with the faster functional iteration $X_{n+1} = A_0(I - \sum_{i=1}^{+\infty} X_n^{i-1} A_i)^{-1}$ [14] and $X_0 = I$ is 207.3 seconds, with 2943 iterations. It is worth pointing out the substantial reduction of the number of iterations (and of the total time) when h passes from 1 to 2, and the strong advantage with respect to the faster functional iteration method starting with the identity matrix.

For Problem 2 we have computed the eigenvalues of the matrix G, by approximating the eigenvalues of an auxiliary $2k \times 2k$ matrix, as suggested in [6], and the eigenvectors, by solving the linear systems (14). The time needed was 0.01 seconds. The eigenvalues, which are real and distinct for this problem, are

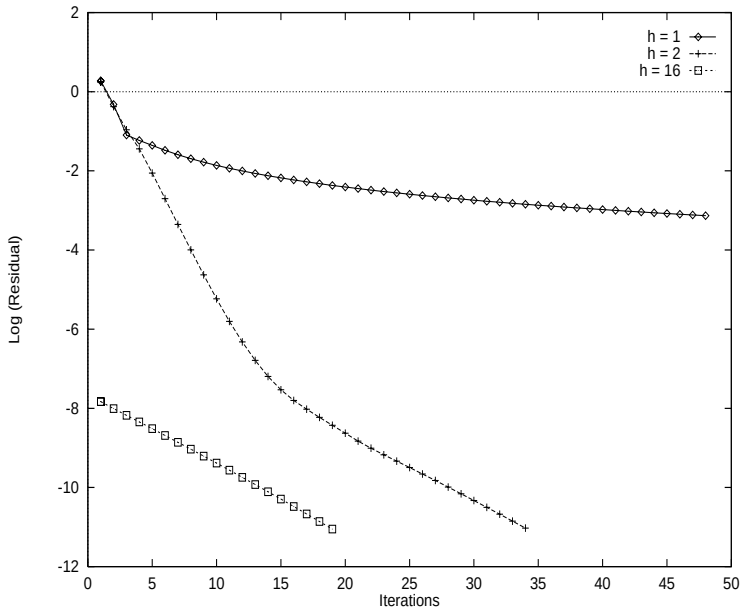


Figure 1: Metaring MAC Protocol

sorted in such a way that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_k|$. We have chosen X_0 with eigenvalues $\lambda_1, \ldots, \lambda_h$, and left eigenvectors $\mathbf{w}_1^t, \ldots, \mathbf{w}_h^t$, for different values of h in $\{1, \ldots, k\}$. Table 2 reports: the number h of eigenvalues shared by X_0 with G; the number of iterations needed by the functional iteration method to reach the required accuracy; the total time (in seconds) needed to compute the matrix X_0 and to perform the functional iterations.

From Table 2 and Figure 2, we can observe a substantial reduction of the number of iterations for increasing values of h. When $h = 24$, i.e., when all the eigenvalues and the corresponding left eigenvectors are known, the matrix X_0 is a poor approximation of G, that is refined in few steps of functional iteration.

References

[1] O. Aberth. Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comp.*, 27, 1973.

[2] G. Anastasi, L. Lenzini, and B. Meini. Performance evaluation of a worst case model of the MetaRing MAC protocol with global fairness. *Performance Evaluation*, 29:127–151, 1997.

h	Iterations	Time (s.)
1	83361	167.1
2	67271	134.1
5	39732	79.8
10	16718	33.5
12	11198	24.0
15	4736	9.9
18	340	0.8
24	340	0.8

Table 2: PH/PH/D Queue

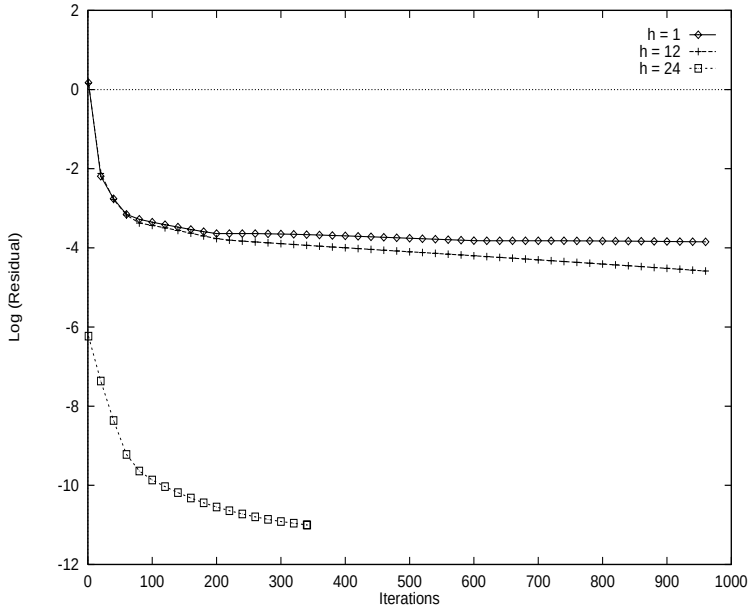


Figure 2: PH/PH/D Queue

[3] D. Bini and B. Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 21–38. Kluwer Academic Publisher, 1995.

[4] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:906–926, 1996.

[5] D. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 15:57–74, 1997.

[6] J. N. Daigle and D. M. Lucantoni. Queueing systems having phase-dependent arrival and service rates. In W. J. Stewart, editor, *Numerical solution of Markov Chains*, pages 161–202. Marcel Dekker, New York, 1991.

[7] L. W. Ehrilch. A modified Newton method for polynomials. *Comm. A.C.M.*, 10:107–108, 1967.

[8] P. Favati and B. Meini. Relaxed functional iteration techniques for the numerical solution of M/G/1 type Markov chains. *BIT*, 38, 1998. In printing.

[9] 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*, 10:1–43, 1994.

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

[11] G. Latouche. Newton's iteration for non-linear equations in Markov chains. *IMA J. of Numerical Analysis*, 14:583–598, 1994.

[12] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Probability*, 30:650–674, 1993.

[13] G. Latouche and G.W. Stewart. Numerical methods for M/G/1 type queues. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 571–581. Kluwer Academic Publishers, 1996.

[14] B. Meini. New convergence results on functional iteration techniques for the numerical solution of M/G/1 type Markov chains. *Numer. Math.*, 78:39–58, 1997.

[15] G. R. Murthy, M. Kim, and E. J. Coyle. Equilibrium analysis of skip free Markov chains: nonlinear matrix equations. *Commun. Statist. Stochastic Models*, 7:547–571, 1991.

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

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

[18] R. S. Varga. *Matrix Iterative Analysis*. Prentice Hall, New Jersey, 1962.