Exploiting the Toeplitz Structure in Certain Queueing Problems

Dario Andrea Bini and Beatrice Meini
Dipartimento di Matematica, Università di Pisa
bini@dm.unipi.it, meini@dm.unipi.it

Abstract

Toeplitz matrices model a wide variety of problems in queueing theory. However, the algorithmic tools used in this field do not fully exploit the Toeplitz structure. In this paper, besides presenting a simple example that points out the role of Toeplitz matrices in queueing theory, we describe very recent results where some customary concepts, as displacement rank, Schur complement, FFT and fast polynomial computations, are exploited in order to devise very efficient algorithms for solving a wide class of queueing problems.

1 Introduction

A wide variety of problems in queueing theory can be modelled by Markov chains defined by stochastic (block) Toeplitz matrices having the Hessenberg structure. This occurs, say, whenever the probability of transition between state i to state j just depends on the difference $i-j$. This feature is only partially and rarely used in the currently available algorithms for solving queueing problems (compare [14], [24]).

In this paper we report some recent advances in devising numerical methods for Markov chains where the specific tools of the Toeplitz matrix technology are used in order to dramatically increase the efficiency of the algorithms.

More specifically, we show that the well known concepts related to Toeplitz matrices, as displacement rank, Schur complement, and the correlations intercurring between Toeplitz matrices and polynomials, allow us, on one hand to improve existing algorithms for solving Markov chains, on the other hand to devise new and more efficient specific algorithms that dramatically reduce the cpu time needed for solving a large class of queueing problems and Markov chains.

Here we present some recent results [4], [5], [6], [27], [7] concerning the improvement of the Latouche-Stewart algorithm [24], a new fast version of the well known Ramaswami formula [32], and a new algorithm based on a functional formulation of the cyclic reduction method.

The paper is organized as follows: in section 2 we motivate the role of Toeplitz matrices by showing a simple example concerning the shortest queue problem. In section 3 we recall the main computational problems encountered in the numerical solution of Markov chains. In section 4 we recall the main tools for Toeplitz computations. Finally in section 5 we present our algorithms for solving Markov chains that fully exploit the Toeplitz structure.

2 Toeplitz matrices and queues

A very simple example that points out the role of Toeplitz matrices in queueing theory is the shortest queue problem. Consider a system with $k \geq 2$ servers, each with its own queue with infinite capacity. Customers arrive according to a Poisson process with rate λ and, immediately upon arrival, join the shortest queue. More precisely, the probability that i customers arrive in a single unit of time is $a(i) = e^{-\lambda}\frac{\lambda^i}{i!}$. If there is not a queue that is shorter all other queues, a customer joins any of the shortest queues. Customers are served on a first-come first-served basis, with no jockeying or defection allowed. The service times are constant of unitary length and the services are synchronized: we suppose that each service begins and ends at integer instants of time.

For the symmetry of the system the situation is fully determined by the total number of customers who are in the system at any integer instant of time. Therefore the problem consists in estimating the probability π_j of having a total number j of customers waiting to be served, for $j = 0, 1, 2, \ldots$, when the time tends to infinity.

The vector $\boldsymbol{\pi} = (\pi_j)$ is called probability invariant vector and is such that $\sum_j \pi_j = 1$, $P\boldsymbol{\pi} = \boldsymbol{\pi}$, where $P = (p_{i,j})$ is the transition probability matrix, i.e., $p_{i,j}$ is the probability that at the general instant of time the total number of customers waiting in the queues changes from j to i.

It is immediate to verify that

$$p_{i,j} = \begin{cases} a(i-j+k) & \text{for } j \geq k,\ i-j+k \geq 0, \\ 0 & \text{for } j \geq k,\ i-j+k < 0, \\ a(i) & \text{for } j < k, \end{cases}$$

that is,

$$P = \left[\begin{array}{cccccc} \multicolumn{3}{c}{\overbrace{\hphantom{a_0 \ \ldots \ a_0}}^{k}} & & & \\ a_0 & \ldots & a_0 & & & \bigcirc \\ a_1 & \ldots & a_1 & a_0 & & \\ a_2 & \ldots & a_2 & a_1 & a_0 & \\ \vdots & & \vdots & \ddots & \ddots & \ddots \end{array} \right],$$

where P is column stochastic, i.e., $\sum_i p_{i,j} = 1$, $j = 0, 1, \ldots$.

The above matrix, except for its first $k-1$ columns, is a Toeplitz matrix. We may partition P into $m \times m$ blocks, $m = k-1$, and obtain the following block Toeplitz-like matrix in block Hessenberg form

$$P = \begin{bmatrix} B_1 & A_0 & & & \bigcirc \\ B_2 & A_1 & A_0 & & \\ B_3 & A_2 & A_1 & A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}. \qquad (1)$$

Stochastic matrices of the structure (1) are known in the literature of Markov chains as matrices of the M/G/1 type [30]. They model many applications where the blocks A_i and B_i may be any $m \times m$ nonnegative matrices or may have specific sparsity patterns or structures, moreover they may occur with finite or with infinite size [14], [2], [25], [1], [30], [29], [21], [26], [11]. Similar problems are modelled by G/M/1 type matrices, i.e., column stochastic matrices having the block Toeplitz-like block Hessenberg structure of P^T [29].

3 Computational problems

The central problem in queueing analysis is computing the solution of the homogeneous linear system

$$(I-P)\boldsymbol{\pi} = \mathbf{0}, \quad \sum_i \pi_i = 1, \qquad (2)$$

where the matrix P is infinite or has a very large size and has the structure (1).

Conditions for the existence of the solution of (2) are given in [30].

The large matrix sizes involved in the applications make very important to exploit all the specific features of P in order to devise efficient algorithms for solving (2) endowed with a low computational cost.

Many existing algorithms are based on the solution of (2) by means of direct or iterative techniques. Direct methods, mainly based on LU decomposition, may be applied for matrices of finite size [12], [13], [16]. Iterative methods, needed for infinite matrices, can be used also in the finite case [4], [20].

A classical method used for solving (2), when the matrix P is infinite, is the well-known Ramaswami formula [32], that allows the recursive computation of the components of the vector $\boldsymbol{\pi}$, once the solution of a nonlinear matrix equation is known. More precisely, let G be the $m \times m$ matrix given by the minimal nonnegative solution of the nonlinear equation

$$X = \sum_{i=0}^{\infty} X^i A_i, \qquad (3)$$

and let partition the vector $\boldsymbol{\pi}$ as $\boldsymbol{\pi}^T = [\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \ldots]^T$, according to the block structure of (1), where the $\boldsymbol{\pi}_i$'s have m components. Then the following relations hold (Ramaswami's formula [32]):

$$\begin{array}{l} \bar{B}_0\boldsymbol{\pi}_0 = \boldsymbol{\pi}_0, \quad \mathbf{k}^T\boldsymbol{\pi}_0 = 1 \\ \boldsymbol{\pi}_i = (I - \bar{A}_1)^{-1}\left[\bar{B}_i\boldsymbol{\pi}_0 + \sum_{j=1}^{i-1}\bar{A}_{i+1-j}\boldsymbol{\pi}_j\right], \; i \geq 1, \end{array} \tag{4}$$

where

$$\begin{array}{l} \bar{A}_{i+1} = \sum_{j=i+1}^{+\infty} G^{j-i-1}A_j, \quad \bar{B}_i = \sum_{j=i}^{+\infty} G^{j-i}B_j, \quad i \geq 0, \\ \mathbf{k}^T = \mathbf{e}^T + \mathbf{e}^T(I - \sum_{i=1}^{+\infty}\bar{A}_i)^{-1}. \end{array} \tag{5}$$

It is extremely important to devise and analyze fast and efficient methods for the solution of the matrix equation (3). Several techniques have been introduced in the recent literature: functional iterations having linear convergence have been analyzed in [21], [31], [17], [28]; doubling techniques have been introduced in [33], [24]; Newton's iterations are analyzed in [22]; a logarithmic reduction algorithm is proposed in [23].

Another way for solving (3) is to rewrite it as a following block Toeplitz block Hessenberg infinite system:

$$[G, G^2, G^3, \ldots]\begin{bmatrix} I - A_1 & -A_0 & & & O \\ -A_2 & I - A_1 & -A_0 & & \\ -A_3 & -A_2 & I - A_1 & -A_0 & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} = [A_0, O, O, \ldots]. \tag{6}$$

By using this formulation, a dramatic reduction of the complexity has been achieved by using the Toeplitz matrix technology: cyclic reduction has been succesfully applied in functional form in [4], [5], [6]; the properties of displacement rank have been exploited in [7].

4 Toeplitz matrix tools

In this section we recall the basic tools used for devising fast algorithms for the solution of problems (2) and (3). More precisely we recall the FFT-based techniques for implementing fast matrix polynomial arithmetic and relate them to Toeplitz computations. Then we recall some basic concepts concerning displacement operators and displacement rank.

4.1 Matrix polynomial arithmetic

Matrix polynomials, matrix power series and block Toeplitz (infinite) matrices are intimately related. The formal relation intercurring among the coefficients of the matrix power series $A(z) = \sum_{i=0}^{+\infty} A_i z^i$, $B(z) = \sum_{i=0}^{+\infty} B_i z^i$, $C(z) = \sum_{i=0}^{+\infty} C_i z^i$, such that $C(z) = A(z)B(z)$ can be easily rewritten in matrix form as

$$\begin{bmatrix} C_0 & & & O \\ C_1 & C_0 & & \\ C_2 & C_1 & C_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} A_0 & & & O \\ A_1 & A_0 & & \\ A_2 & A_1 & A_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} B_0 & & & O \\ B_1 & B_0 & & \\ B_2 & B_1 & B_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

In this way block Toeplitz matrix computations can be reduced to manipulating formal power series (for infinite matrices) or polynomials (for finite matrices). Moreover, power series that are convergent in the unit circle can be numerically reduced to polynomials by neglecting the block coefficients which are smaller than the working precision. In this way the above matrix equation can be reduced to a finite size.

Due to this correlation, we may use the technique of evaluation/interpolation at the roots of 1, based on FFT, for performing fast Toeplitz computations.

In particular, we introduce an algorithm for matrix polynomial multiplication modulo $z^n - 1$, and then use it for the computation of the product of a block Toeplitz matrix and a block vector. Then we reduce inversion of a matrix polynomial modulo z^n to inverting a lower block triangular block Toeplitz matrix, and we introduce an algorithm for this computation.

Throughout this section we will denote $\mathbf{y} = (y_0, \ldots, y_{n-1}) = \mathrm{DFT}(\mathbf{x})$ the Discrete Fourier Transform of the vector $\mathbf{x} = (x_0, \ldots, x_{n-1})$ defined by $y_i = \sum_{j=0}^{n-1} \omega^{ij} x_j$, $i = 0, \ldots, n-1$, where $\omega = \cos\frac{2\pi}{d} + \mathbf{i}\sin\frac{2\pi}{d}$, and $\mathbf{i}$ is the imaginary unit such that $\mathbf{i}^2 = -1$. Analogously $\mathbf{x} = \mathrm{IDFT}(\mathbf{y})$ denotes the Inverse Discrete Fourier Transform such that $\mathbf{x} = \frac{1}{n}\sum_{j=0}^{n-1} \omega^{-ij} y_j$, $i = 0, \ldots, n-1$. It is well known that if $n = 2^M$, where M is a positive integer, the computation of DFT and IDFT can be performed with $O(n \log n)$ real arithmetic operations (hereafter denoted by ops) by means of the base-2 FFT algorithms.

Similarly, we extend the above notation to block vectors. Given the block vector $X = (X_0, \ldots, X_{n-1})$, where $X_s = (x_{i,j}^{(s)})$ are $m \times m$ matrices, we define $Y = (Y_0, \ldots, Y_{n-1}) = \mathrm{DFT}(X)$ the block vector such that $Y_s = (y_{i,j}^{(s)})$ are $m \times m$ matrices and $(y_{i,j}^{(0)}, \ldots, y_{i,j}^{(n-1)}) = \mathrm{DFT}(x_{i,j}^{(0)}, \ldots, x_{i,j}^{(n-1)})$, $i, j = 1, \ldots, m$. Analogously we set $X = \mathrm{IDFT}(Y)$ if $(x_{i,j}^{(0)}, \ldots, x_{i,j}^{(n-1)}) = \mathrm{IDFT}(y_{i,j}^{(0)}, \ldots, y_{i,j}^{(n-1)})$. In this way the computation of $\mathrm{DFT}(X)$ and $\mathrm{IDFT}(Y)$ is reduced to the computation of m^2 DFT's and IDFT's, respectively, for the cost of $O(m^2 n \log n)$ ops.

Let $P(z) = \sum_{i=0}^{n-1} P_i z^i$, $Q(z) = \sum_{i=0}^{n-1} Q_i z^i$, $R(z) = \sum_{i=0}^{n-1} R_i z^i$ be matrix polynomials such that $R(z) = P(z)Q(z) \bmod (z^n - 1)$. Then the matrix coefficients of $R(z)$ can be computed, given the matrix coefficients of $P(z)$ and $Q(z)$, by means of the following algorithm based on evaluation/interpolation at the roots of 1.

Algorithm 1. *Computation of matrix polynomial product modulo $z^n - 1$.*

Input. Positive integers M, m and the $m \times m$ matrices $P_0, \ldots, P_{n-1}$, $Q_0, \ldots, Q_{n-1}$, coefficients of the matrix polynomials $P(z) = \sum_{i=0}^{n-1} P_i z^i$, $Q(z) = \sum_{i=0}^{n-1} Q_i z^i$, where $n = 2^M$.

Output. The $m \times m$ matrices $R_0, \ldots, R_{n-1}$, coefficients of the matrix polynomial $R(z) = \sum_{i=0}^{n-1} R_i z^i$ such that $R(z) = P(z)Q(z) \bmod (z^n - 1)$.

Computation.

1. (Evaluation) Compute the entries of the $2n$ matrices $U_s = P(\omega^s)$, $V_s = Q(\omega^s)$, $s = 0, \ldots, n-1$, $\omega = \cos\frac{2\pi}{n} + \mathbf{i}\sin\frac{2\pi}{n}$, in the following way

$$\begin{aligned}(U_0, \ldots, U_{n-1}) &= \mathrm{DFT}(P_0, \ldots, P_{n-1}),\\ (V_0, \ldots, V_{n-1}) &= \mathrm{DFT}(Q_0, \ldots, Q_{n-1}).\end{aligned}$$

2. Compute the n matrix products $W_s = U_s V_s$, $s = 0, \ldots, n-1$, such that $W_s = P(\omega^s)Q(\omega^s) = R(\omega^s)$.

3. (Interpolation) Compute the entries of the matrices $R_0, \ldots, R_{n-1}$ by means of the equation $(R_0, \ldots, R_{n-1}) = \mathrm{IDFT}(W_0, \ldots, W_{n-1})$.

The cost of Algorithm 1 is $O(m^3 n + m^2 n \log n)$ ops.

Remark 1 Observe that if the input polynomials $P(z)$ and $Q(z)$ have degree at most $n/2 - 1$ then $R(z) = P(z)Q(z)$, thus, matrix polynomial product can be computed by means of Algorithm 1.

Remark 2 Algorithm 1 can be also used for efficiently computing the product of a block Toeplitz matrix and a vector. In fact, if $P(z)$ has degree $n-1$ and $Q(z)$ has degree $n/2-1$, then, comparing the terms of degree $n/2, \ldots, n-1$ in the equation $R(z) = P(z)Q(z) \bmod (z^n - 1)$, we arrive at the following equation involving a block Toeplitz matrix

$$\begin{bmatrix} R_{n/2} \\ \vdots \\ R_{n-1} \end{bmatrix} = \begin{bmatrix} P_{n/2} & \ldots & P_1 \\ \vdots & \ddots & \vdots \\ P_{n-1} & \ldots & P_{n/2} \end{bmatrix} \begin{bmatrix} Q_0 \\ \vdots \\ Q_{n/2-1} \end{bmatrix}.$$

Comparing the coefficients of the term of degree i in both sides of the equation $P(z)Q(z) = I \bmod z^n$ yields the matrix equation involving a lower block triangular block Toeplitz matrix

$$\begin{bmatrix} P_0 & & & \bigcirc \\ P_1 & P_0 & & \\ \vdots & \ddots & \ddots & \\ P_{n-1} & \ldots & P_1 & P_0 \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{n-1} \end{bmatrix} = \begin{bmatrix} I \\ O \\ \vdots \\ O \end{bmatrix} \tag{7}$$

Let us denote by T_n the matrix in (7), assume for simplicity $n = 2^M$, M positive integer, and partition T_n as follows

$$T_n = \begin{bmatrix} T_{n/2} & O \\ H_{n/2} & T_{n/2} \end{bmatrix}$$

where all the four blocks are $(n/2) \times (n/2)$ block Toeplitz matrices. Then the matrix T_n^{-1} can be easily written as

$$T_n^{-1} = \begin{bmatrix} T_{n/2}^{-1} & O \\ -T_{n/2}^{-1} H_{n/2} T_{n/2}^{-1} & T_{n/2}^{-1} \end{bmatrix} \tag{8}$$

and the first block column $(Q_0^T, \ldots, Q_{n-1}^T)^T$ of T_n^{-1}, which solves (7), can be computed from the first block column $(Q_0^T, \ldots, Q_{n/2-1}^T)^T$ of $T_{n/2}^{-1}$ by means of two multiplications between an $(n/2) \times (n/2)$ block Toeplitz matrix and a block vector. In this way we arrive at the following algorithm which extends to the block case the algorithm of Lafon and Sieveking-Kung (compare [9]).

Algorithm 2. *Solution of the congruence* $P(z)Q(z) = I \bmod z^n$, *or, equivalently, of the system (7).*

Input. A positive integer M, the $m \times m$ matrices $P_0, P_1, \ldots, P_{n-1}$, $n = 2^M$, defining the matrix polynomial $P(z) = \sum_{i=0}^{n-1} P_i z^i$, $\det P_0 \neq 0$.

Output. The matrices $Q_0, Q_1, \ldots, Q_{n-1}$ satisfying (7), or equivalently, such that the polynomial $Q(z) = \sum_{i=0}^{n-1} Q_i z^i$ solves the congruence $P(z)Q(z) = I \bmod z^n$.

Computation.

1. Compute $Q_0 = P_0^{-1}$.

2. For $i = 0, \ldots, M-1$, given the first column $U = (Q_0^T, \ldots, Q_{2^i-1}^T)^T$ of $T_{2^i}^{-1}$, compute the block vector $V = (Q_{2^i}^T, \ldots, Q_{2^{i+1}-1}^T)^T$, which defines the remaining blocks of the first column of $T_{2^{i+1}}^{-1}$, by applying equation (8) with $n = 2^{i+1}$ in the following way: by applying Algorithm 1 and Remark 2 compute the products $W = H_{2^i} U$ and $V = T_{2^i}^{-1} W$.

The cost of Algorithm 2 is $O(m^3 n + m^2 n \log n)$ ops.

4.2 Displacement rank

The concept of displacement rank, introduced by Kailath et al. in [19], is fundamental in devising and analyzing algorithms related to Toeplitz matrices. An excellent survey on the properties and the applications of the displacement rank, together with a very rich bibliography, is presented by Kailath and Sayed in [18]. Applications to parallel computations, polynomial computations and Toeplitz/Hankel-like matrices can be found in [9], [8], [3]. Let us recall some properties of displacement operators which are particularly suitable for dealing with the inversion of a block Toeplitz matrix in block Hessenberg form.

Define the $n \times n$ *block down-shift* matrix

$$Z_n = \begin{bmatrix} O & & & \\ I & O & & \\ & \ddots & \ddots & \\ & & I & O \end{bmatrix},$$

where the blocks have dimension m, and consider the following *block displacement operator*

$$\Delta(H) = Z_n H - H Z_n$$

defined for any $n \times n$ block matrix H.

Let us also introduce the following notations: the block vectors $E_n^{(1)} = [I, O, \ldots, O]^T$, $E_n^{(2)} = [O, \ldots, O, I]^T$ denote $mn \times m$ matrices made up by the first and the last m columns of the identity matrix of order mn. Throughout we call n-dimensional *block column vector* any $mn \times m$ matrix, similarly we call n-dimensional *block row vector* any $m \times mn$ matrix. Moreover we say that the block matrix H has block displacement rank r if r is the minimum integer such that there exist block column vectors $U^{(i)}$, and block row vectors $V^{(i)}$, $i = 1, \ldots, r$, satisfying the equation $\Delta(H) = \sum_{i=1}^{r} U^{(i)} V^{(i)}$.

A nice property of $\Delta(H)$ is that the block displacement of the inverse matrix can be explicitely related to the block displacement of the matrix itself, in fact, if H is nonsingular, then

$$\Delta(H^{-1}) = -H^{-1} \Delta(H) H^{-1}. \tag{9}$$

The following result can be easily proved by extending to block matrices the same proof given in [9].

Theorem 4.1 *Let K_n be an $n \times n$ block matrix such that $\Delta(K_n) = \sum_{i=1}^{r} U^{(i)} V^{(i)}$ where $U^{(i)}$ and $V^{(i)}$ are n-dimensional block column and block row vectors, respectively. Then we have*

$$K_n = L_n(K_n E_n^{(1)}) - \sum_{i=1}^{r} L_n(U^{(i)}) L_n^T(Z_n V^{(i)T})$$

where $L_n(W)$ denotes the $n \times n$ block lower triangular block Toeplitz matrix defined by its first block column W.

The above result allows one to represent any matrix K_n as a sum of products of block lower and upper triangular block Toeplitz matrices defined by the first block column of K_n and by the block vectors $U^{(i)}$, $V^{(i)}$ associated with the block displacement of K_n.

If the matrix K_n is nonsingular then the above representation theorem can be applied to K_n^{-1} in the light of (9) yielding the following result

$$K_n^{-1} = L_n(K_n^{-1}E_n^{(1)}) + \sum_{i=1}^{r} L_n(K_n^{-1}U^{(i)})L_n^T(Z_n K_n^{-T}V^{(i)T}). \tag{10}$$

5 Solution of the computational problems

In this section we use the tools introduced in section 4 in order to devise fast algorithms for solving problems (2), (3). We first consider the Ramaswami formula (4), then we use displacement operators for problem (3), finally we apply fast Toeplitz and polynomial computations for devising an algorithm, based on cyclic reduction, which solves (2), (3).

5.1 The fast Ramaswami formula

By means of the natural isomorphism between formal power series and triangular Toeplitz matrices outlined in section 4.1, it is possible to give a novel interpretation of Ramaswami's formula and to devise a new fast algorithm, based on the use of FFT techniques, for its computation [27].

More precisely, it is possible to show that the block Hessenberg matrix

$$H = \begin{bmatrix} I - A_1 & -A_0 & & \bigcirc & \\ -A_2 & I - A_1 & -A_0 & & \\ -A_3 & -A_2 & I - A_1 & -A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} \tag{11}$$

can be factorized as the product $H = UL$, where

$$U = \begin{bmatrix} I & -G & & & \bigcirc \\ & I & -G & & \\ & & I & -G & \\ \bigcirc & & & \ddots & \ddots \end{bmatrix} \quad L = \begin{bmatrix} I - \bar{A}_1 & & & \bigcirc \\ -\bar{A}_2 & I - \bar{A}_1 & & \\ -\bar{A}_3 & -\bar{A}_2 & I - \bar{A}_1 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

G is the solution of (3) and the $\bar{A}_i$'s are defined in (5). Ramaswami's formula (4) can now be derived by solving the system

$$UL \begin{bmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \boldsymbol{\pi}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \vdots \end{bmatrix} \boldsymbol{\pi}_0.$$

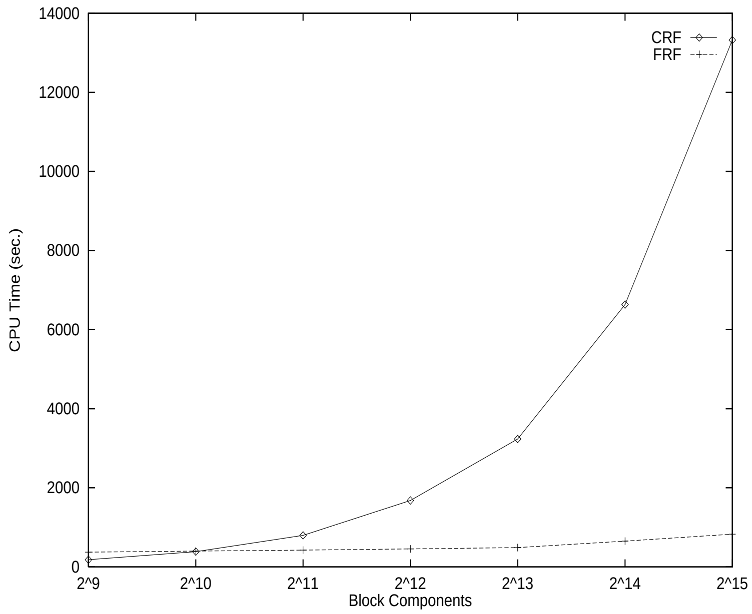

Figure 1: Fast and Customary Ramaswami formula

In fact, by multiplying the above equation by the left inverse of U, we obtain that

$$L \begin{bmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \boldsymbol{\pi}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \\ \bar{B}_3 \\ \vdots \end{bmatrix} \boldsymbol{\pi}_0. \tag{12}$$

Hence, by solving the block lower triangular system (12) by forward substitution, we arrive at (4).

The computation of the vector $\boldsymbol{\pi}$ is so reduced to solving a block lower triangular block Toeplitz infinite system. In [27] a fast Ramaswami formula, based on Algorithms 1 and 2, has been devised with a substantial reduction of the asymptotic computational cost. More precisely, the computational cost to calculate p block components of the probability invariant vector $\boldsymbol{\pi}$ is roughly $p(7m^2 + 40\log n_0)$ arithmetic operations, where n_0 is such that the blocks A_i are negligible for $i > n_0$. The computational cost of the customary Ramaswami formula is $2pn_0m^2$ arithmetic operations. Figure 1 reports the CPU time needed by fast and customary Ramaswami formula applied to a concrete problem of metropolitan networks [2].

5.2 Solving a matrix equation by means of displacement operators

Consider the $n \times n$ block Toeplitz matrix in block Hessenberg form

$$H_n = \begin{bmatrix} I - A_1 & -A_0 & & & \bigcirc \\ -A_2 & I - A_1 & -A_0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ -A_{n-1} & \ddots & \ddots & I - A_1 & -A_0 \\ -A_n & -A_{n-1} & \dots & -A_2 & I - A_1 \end{bmatrix} \tag{13}$$

obtained by truncating the infinite matrix (11). Let us assume for simplicity that $n = 2^q$, for a positive integer q. Suppose that $\det H_{2^j} \neq 0$, $j = 0, 1, \dots, q$, and partition the matrix H_n in the following way

$$H_n = \begin{bmatrix} H_{\frac{n}{2}} & U_{\frac{n}{2}} V_{\frac{n}{2}}^T \\ T_{\frac{n}{2}} & H_{\frac{n}{2}} \end{bmatrix}$$

where

$$U_{\frac{n}{2}} = \begin{bmatrix} O \\ \vdots \\ O \\ I \end{bmatrix}, V_{\frac{n}{2}} = \begin{bmatrix} -A_0^T \\ O \\ \vdots \\ O \end{bmatrix}, \quad T_{\frac{n}{2}} = \begin{bmatrix} -A_{\frac{n}{2}+1} & -A_{\frac{n}{2}} & \dots & -A_2 \\ -A_{\frac{n}{2}+2} & -A_{\frac{n}{2}+1} & \ddots & -A_3 \\ \vdots & \ddots & \ddots & \vdots \\ -A_n & -A_{n-1} & \dots & -A_{\frac{n}{2}+1} \end{bmatrix}$$

and I denotes the $m \times m$ identity matrix. By applying the Sherman Woodbury Morrison formula [15] to the decomposition

$$H_n = \begin{bmatrix} H_{\frac{n}{2}} & O \\ T_{\frac{n}{2}} & H_{\frac{n}{2}} \end{bmatrix} + \begin{bmatrix} O & U_{\frac{n}{2}} V_{\frac{n}{2}}^T \\ O & O \end{bmatrix} = S_n + M_n N_n^T,$$
$$M_n = \begin{bmatrix} U_{\frac{n}{2}} \\ O \end{bmatrix}, N_n = \begin{bmatrix} O \\ V_{\frac{n}{2}} \end{bmatrix}$$

we immediately find the following expression for the matrix inverse of H_n:

$$H_n^{-1} = S_n^{-1} - S_n^{-1} M_n (I + N_n^T S_n^{-1} M_n)^{-1} N_n^T S_n^{-1},$$

where

$$S_n^{-1} = \begin{bmatrix} H_{\frac{n}{2}}^{-1} & O \\ -H_{\frac{n}{2}}^{-1} T_{\frac{n}{2}} H_{\frac{n}{2}}^{-1} & H_{\frac{n}{2}}^{-1} \end{bmatrix}.$$

The above relations have been used in [24], [33] in order to devise a doubling algorithm for the solution of the linear system (6), truncated at a finite system of block dimension n, in $O(m^3 n \log^2 n)$ operations. This algorithm can be much simplified by applying the tool of displacement rank of section 4.2.

In fact, observe that for the matrix H_n of (13) it holds

$$\Delta(H_n) = -\begin{bmatrix} I \\ O \\ \vdots \\ O \end{bmatrix} [A_0, O, \ldots, O] + \begin{bmatrix} O \\ \vdots \\ O \\ I \end{bmatrix} [O, \ldots, O, A_0]. \tag{14}$$

Whence the equation (10) can be much simplified: due to the special structure (14) of $\Delta(H_n)$ we may easily prove that

$$H_n^{-1} = L_n(C_n^{(1)})L_n^T(E_n^{(1)} - Z_n R_n^{(1)T} A_0^T) + L_n(C_n^{(2)})L_n^T(Z_n R_n^{(2)T} A_0^T) \tag{15}$$

where $C_n^{(1)}$, $C_n^{(2)}$ denote the first and the last block column, respectively, of H_n^{-1}, while $R_n^{(1)}$, $R_n^{(2)}$ denote the first and the last block row, respectively, of the matrix H_n^{-1}.

In this way the inverse of H_n is explicitely determined as a sum of products of block triangular block Toeplitz matrices defined by the block A_0, by the block rows $R_n^{(1)}$, $R_n^{(2)}$, and the block columns $C_n^{(1)}$, $C_n^{(2)}$.

The Sherman-Morrison-Woodbury formula allows us to relate the block vectors $C_n^{(1)}$, $C_n^{(2)}$, $R_n^{(1)}$, $R_n^{(2)}$ defining the matrix H_n^{-1} and the block vectors $C_{\frac{n}{2}}^{(1)}$, $C_{\frac{n}{2}}^{(2)}$, $R_{\frac{n}{2}}^{(1)}$, $R_{\frac{n}{2}}^{(2)}$ defining the matrix $H_{\frac{n}{2}}^{-1}$. Moreover, since such relations only involve operations between block Toeplitz matrices and block vectors, we may devise an efficient scheme for their implementation based on Algorithm 1 of section 4.1. In this way we arrive at an algorithm for the computation of the inverses of H_{2^i}, $i = 0, 1, \ldots, q$, requiring $O(m^3 n + m^2 n \log n)$ arithmetic operations, for $n = 2^q$, and a low storage cost; in fact, only four auxiliary block vectors need to be allocated in order to carry out our algorithm. Another interesting feature is that in the PRAM model of parallel computation the algorithm can be carried out in $O(\log^2 n)$ parallel steps using $O(m^3 n)$ processors.

The following result provides the desired correlation intercurring among block rows and columns of H_n^{-1} and $H_{\frac{n}{2}}^{-1}$.

Theorem 5.1 *The first and the last block columns $C_n^{(1)}$, $C_n^{(2)}$, and the first and the last block rows $R_n^{(1)}$, $R_n^{(2)}$ of H_n^{-1} satisfy the following relations*

$$C_n^{(1)} = \begin{bmatrix} C_{\frac{n}{2}}^{(1)} \\ -H_{\frac{n}{2}}^{-1} T_{\frac{n}{2}} C_{\frac{n}{2}}^{(1)} \end{bmatrix} + \begin{bmatrix} C_{\frac{n}{2}}^{(2)} \\ -H_{\frac{n}{2}}^{-1} T_{\frac{n}{2}} C_{\frac{n}{2}}^{(2)} \end{bmatrix} P R_{\frac{n}{2}}^{(1)} T_{\frac{n}{2}} C_{\frac{n}{2}}^{(1)}$$

$$C_n^{(2)} = \begin{bmatrix} O \\ C_{\frac{n}{2}}^{(2)} \end{bmatrix} - \begin{bmatrix} C_{\frac{n}{2}}^{(2)} \\ -H_{\frac{n}{2}}^{-1} T_{\frac{n}{2}} C_{\frac{n}{2}}^{(2)} \end{bmatrix} P R_{\frac{n}{2}}^{(1)} E_{\frac{n}{2}}^{(2)}$$

$$R_n^{(1)} = -A_0[R_{\frac{n}{2}}^{(1)}, O] - A_0 R_{\frac{n}{2}}^{(1)} E_{\frac{n}{2}}^{(2)} P [R_{\frac{n}{2}}^{(1)} T_{\frac{n}{2}} H_{\frac{n}{2}}^{-1}, -R_{\frac{n}{2}}^{(1)}]$$

$$R_n^{(2)} = A_0[R_{\frac{n}{2}}^{(2)} T_{\frac{n}{2}} H_{\frac{n}{2}}^{-1}, -R_{\frac{n}{2}}^{(2)}] + A_0 R_{\frac{n}{2}}^{(2)} T_{\frac{n}{2}} C_{\frac{n}{2}}^{(2)} P [R_{\frac{n}{2}}^{(1)} T_{\frac{n}{2}} H_{\frac{n}{2}}^{-1}, -R_{\frac{n}{2}}^{(1)}],$$

where $P = (I - A_0 R^{(1)}_{\frac{n}{2}} T_{\frac{n}{2}} C^{(2)}_{\frac{n}{2}})^{-1} A_0$.

Once the block vectors $R_n^{(1)}$, $R_n^{(2)}$, $C_n^{(1)}$, $C_n^{(2)}$ have been computed, the solution $\mathbf{x}$ of a system $H_n\mathbf{x} = \mathbf{b}$, can be obtained by means of (15) and Algorithm 1, leaving unchanged the asymptotical cost $O(m^3 n + m^2 n \log n)$.

The following table compares the cpu time needed by our algorithm and the Latouche Stewart algorithm [24].

Table 1 reports the number of cpu seconds required by our algorithm (algorithm DR: Displacement Rank) for solving the matrix equation (3), together with the residual error, for the values of n ranging from 256 to 4096. It reports also the time required by the Latouche–Stewart algorithm [24] (denoted by LS) to reach the same residual error of our algorithm, and the ratio between the times. The symbol "*" denotes that the algorithm is not able to reach the required accuracy for lack of memory.

DR			LS		
n	time	residual	time	residual	ratio
256	8.7	$2.5 \cdot 10^{-3}$	38.3	$2.5 \cdot 10^{-3}$	4.4
512	18.1	$4.0 \cdot 10^{-4}$	103.5	$4.0 \cdot 10^{-4}$	5.7
1024	39.2	$1.7 \cdot 10^{-5}$	264.9	$1.7 \cdot 10^{-5}$	6.8
2048	89.3	$3.4 \cdot 10^{-8}$	*	*	*
4096	193.4	$1.3 \cdot 10^{-13}$	*	*	*

Table 1: Fast and customary divide-and-conquer algorithm

5.3 Solving a matrix equation by means of cyclic reduction

In this section we propose a method for the numerical solution of both (3) and (2), introduced in [6], [5], [4], which is quadratically convergent, numerically stable and, unlike the method of section 5.2, avoids the need of cutting the infinite matrix (6) to a finite size. This method, based on the properties of the Schur complement of block Toeplitz matrices that we describe in functional form, requires a small number of arithmetic operations. In fact, the most expensive computations at each stage of our algorithm are reduced to performing operations between matrix power series, and for these computations we may use the algorithms described in section 4.1.

Our method relies on the technique of the successive state reduction (cyclic reduction [10]), applied for the computation of the vector $\boldsymbol{\pi}$, and for the computation of the solution G of (3). In this section we only consider the problem of computing the solution of (3) and we refer the reader to [4] for problem (2).

Consider the infinite system (6). By performing an odd-even permutation of the block rows and block columns in (6) we find that

$$[G, G^3, \ldots \mid I, G^2, \ldots] \begin{bmatrix} T_1^{(0)} & W^{(0)} \\ Z^{(0)} & T_2^{(0)} \end{bmatrix} = [O, O, \ldots \mid A_0, O, \ldots] \tag{16}$$

where $T_1^{(0)}$, $T_2^{(0)}$ and $Z^{(0)}$ are infinite block triangular block Toeplitz matrices, $W^{(0)}$ is an infinite block Hessenberg block Toeplitz matrix. Applying one step of block-Gaussian elimination to the 2×2 block system (16) yields

$$[I, G^2, G^4, \ldots]H^{(1)} = [A_0, O, O, \ldots] \tag{17}$$

where $H^{(1)} = T_2^{(0)} - Z^{(0)}T_1^{(0)^{-1}}W^{(0)}$ is the Schur complement of $T_2^{(0)}$. It is interesting to observe that $Q^{(1)} = I - H^{(1)}$ is a nonnegative block Hessenberg matrix which, except for the first block column, has the block Toeplitz structure of the matrix of (6).

In other words, $H^{(1)}$ is univocally determined by the blocks $\widehat{A}_i^{(1)}$, $i \geq 1$, $A_i^{(1)}$, $i \geq 0$, defining the first two block columns of $Q^{(1)}$.

We may recursively apply the same reduction (cyclic reduction) to the matrix equation (17) thus obtaining a sequence of matrices $\{H^{(j)}\}$ such that $Q^{(j)} = I - H^{(j)}$ is a nonnegative matrix having, except for the first block column, the same block Hessenberg and Toeplitz structure of the matrix in (6). Each matrix $Q^{(j)}$ is univocally determined by the blocks $\widehat{A}_i^{(j)}$, $i \geq 1$, $A_i^{(j)}$, $i \geq 0$, defining its first two block columns.

The block entries $\{A_i^{(j+1)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(j+1)}\}_{i\geq 1}$, obtained at step $j+1$, are related to the block entries $\{A_i^{(j)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(j)}\}_{i\geq 1}$, obtained at step j, 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 $j \geq 0$, let us associate with the matrix sequences $\{A_i^{(j)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(j)}\}_{i\geq 1}$, the formal matrix power series $\varphi^{(j)}(z) = \sum_{i=0}^{+\infty} A_i^{(j)} z^i$, $\widehat{\varphi}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{i+1}^{(j)} z^i$, respectively, where $A_i^{(0)} = \widehat{A}_i^{(0)} = A_i$, $i \geq 1$, $A_0^{(0)} = A_0$. Then the following functional relations hold:

$$\begin{cases} \varphi^{(j+1)}(z) = z\varphi_{odd}^{(j)}(z) + \varphi_{even}^{(j)}(z)\big(I - \varphi_{odd}^{(j)}(z)\big)^{-1}\varphi_{even}^{(j)}(z) \\ \widehat{\varphi}^{(j+1)}(z) = \widehat{\varphi}_{odd}^{(j)}(z) + \varphi_{even}^{(j)}(z)(I - \varphi_{odd}^{(j)}(z))^{-1}\widehat{\varphi}_{even}^{(j)}(z) \end{cases} \tag{18}$$

where

$$\begin{array}{ll} \varphi_{even}^{(j)}(z) = \sum_{i=0}^{+\infty} A_{2i}^{(j)} z^i, & \varphi_{odd}^{(j)}(z) = \sum_{i=0}^{+\infty} A_{2i+1}^{(j)} z^i \\ \widehat{\varphi}_{even}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{2(i+1)}^{(j)} z^i, & \widehat{\varphi}_{odd}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{2i+1}^{(j)} z^i. \end{array}$$

The relations (18), 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 $\{A_i^{(j)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(j)}\}_{i\geq 1}$, by using the algorithms of section 4.1.

The following results are fundamental for devising an efficient algorithm based on cyclic reduction:

Theorem 5.2 *The blocks* $\{A_i^{(j)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(j)}\}_{i\geq 1}$ *are nonnegative matrices such that* $\sum_{i=0}^{+\infty} A_i^{(j)}$, $A_0 + \sum_{i=1}^{+\infty} \widehat{A}_i^{(j)}$ *are stochastic. Moreover, if the matrix* $I - \sum_{i=0}^{+\infty} G^{i\cdot 2^j} \widehat{A}_{i+1}^{(j)}$ *is nonsingular, then*

$$G = A_0 \left(I - \sum_{i=0}^{+\infty} G^{i\cdot 2^j} \widehat{A}_{i+1}^{(j)} \right)^{-1}.$$

Under mild conditions, usually satisfied in the applications, the block entries $A_i^{(j)}$ and $\widehat{A}_i^{(j)}$ quadratically converge to zero, for $i \geq 2$, for $j \to \infty$, as stated by the following theorem [4], [5], [6]:

Theorem 5.3 *Let* $G' = \lim_{j\to\infty} G^j$. *Then the following convergence properties hold:*

1. $\lim_{j\to\infty} \widehat{A}_i^{(j)} = 0$, *for* $i \geq 2$;
2. *if the entries of the matrix* $(I - \sum_{i=1}^{+\infty} A_i^{(j)})^{-1}$ *are bounded above by a constant, then the sequence of matrices*

$$R^{(j)} = A_0^{(j)} \left(I - \sum_{i=1}^{+\infty} A_i^{(j)} \right)^{-1}$$

   converges quadratically to the matrix G';
3. *if the solution* G *of (3) is irreducible then*

$$\begin{array}{l} \lim_{j\to\infty} A_0^{(j)} (I - A_1^{(j)})^{-1} = G' \\ \lim_{j\to\infty} A_i^{(j)} = 0, \;\; i \geq 2. \end{array}$$

Table 2 reports the CPU time (in seconds) and the residual error needed by the Cyclic Reduction (CR) algorithm, by the algorithm DR of section 5.2 and by the functional iteration method (FI) of [31] to compute an approximation of the matrix G of (3) for two problems arising from modelling metropolitan networks [2].

References

[1] A. S. Alfa. A discrete MAP/PH/1 queue with vacations and exhaustive time-limited service. *Oper. Research Letters*, 18:31–40, 1995.

	CR	DR	FI
Time	1.4	194	6.8
Residual	$5.4 \cdot 10^{-14}$	$1.3 \cdot 10^{-13}$	$4.3 \cdot 10^{-14}$
Time	2.4	194	705
Residual	$4.2 \cdot 10^{-14}$	$1.7 \cdot 10^{-5}$	$1.8 \cdot 10^{-14}$

Table 2: Comparison of CR, DR, FI algorithms

[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.

[3] D. Bini. On a class of matrices related to Toeplitz matrices. Technical Report 83-5, SUNY University, Albany, New York, 1983.

[4] 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, 1996.

[5] 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.

[6] D. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 1997. To appear.

[7] D. Bini and B. Meini. Inverting block Toeplitz matrices in block Hessenberg form by means of displacement operators: application to queueing problems. *Linear Algebra Appl.*, 1997. To appear.

[8] D. Bini and V. Pan. Improved parallel computations with Toeplitz-like and Hankel matrices. *Linear Algebra Appl.*, 188–189:3–29, 1993.

[9] D. Bini and V. Pan. *Matrix and Polynomial Computations, Vol. 1: Fundamental Algorithms.* Birkhäuser, Boston, 1994.

[10] B. L. Buzbee, G. H. Golub, and C. W. Nielson. On direct methods for solving Poisson's equation. *SIAM J. Num. Anal.*, 7:627–656, 1970.

[11] 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.

[12] R. E. Funderlic and R. J. Plemmons. *LU* decomposition of M-matrices by elimination without pivoting. *Linear Algebra Appl.*, 41:99–110, 1981.

[13] R. E. Funderlic and R. J. Plemmons. A combined direct-iterative method for certain M-matrix linear systems. *SIAM J. Alg. Discr. Meth.*, 5:33–42, 1984.

[14] H. R. Gail, S. L. Hantler, and B. A. Taylor. Non-skip-free M/G/1 and G/M/1 type Markov chains. *Adv. Appl. Prob.*, 1997. To appear.

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

[16] 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.

[17] L. Gun. Experimental results on matrix-analytical solution techniques. extensions and comparisons. *Commun. Statist. Stochastic Models*, 5:669–682, 1989.

[18] T. Kailath and A.H. Sayed. Displacement structure: theory and applications. *SIAM Review*, 37:297–386, 1995.

[19] T. Kailath, A. Viera, and M. Morf. Inverses of Toeplitz operators, innovations and orthogonal polynomials. *SIAM Review*, 20:106–119, 1978.

[20] U. R. Krieger. On a two-level multigrid solution method for finite Markov chains. *Linear Algebra Appl.*, 223–224:415–438, 1995.

[21] 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.

[22] G. Latouche. Newton's iteration for non-linear equations in Markov chains. *IMA J. of Numerical Analysis*, 14:583–598, 1994.

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

[24] 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.

[25] L. Lenzini, B. Meini, and E. Mingozzi. An approximate PRMA++ performance and stability analysis. Submitted for publication, 1997.

[26] D. M. Lucantoni and M. F. Neuts. Some steady state distributions for the MAP/SM/1 queue. *Commun. Statist. Stochastic Models*, 10:575–598, 1994.

[27] B. Meini. An improved FFT-based version of Ramaswami's formula. *Commun. Statist. Stochastic Models*, 13:223–238, 1997.

[28] B. Meini. New convergence results on functional iteration techniques for the numerical solution of M/G/1 type Markov chains. *Numer. Math.*, 1997. To appear.

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

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

[31] V. Ramaswami. Nonlinear matrix equations in applied probability - Solution techniques and open problems. *SIAM Review*, 30:256–263, 1988.

[32] 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.

[33] G. W. Stewart. On the solution of block Hessenberg systems. *Numer. Lin. Algebra with Appl.*, 2:287–296, 1995.