Solving M/G/1 type Markov chains: recent advances and applications

Beatrice Meini
Dipartimento di Matematica
Università di Pisa
Via Buonarroti 2
56127 Pisa, Italy
E-mail: meini@dm.unipi.it

Abstract

A survey on numerical methods, based on Toeplitz matrix computation, for the solution of M/G/1 type Markov chains is presented together with some new advances. We show how the block Toeplitz structure, shared by the transition matrices associated with M/G/1 type Markov chains, can be exploited as the basis for devising fast and numerically stable algorithms, for the computation of the probability invariant vector and for the solution of the nonlinear matrix equation $X = \sum_{i=0}^{+\infty} A_i X^i$.

Key words. M/G/1 type matrices, Toeplitz matrices, Fast Fourier Transform.

1 INTRODUCTION

Discrete time Markov chains of M/G/1 type are defined by a transition probability matrix of the form

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

where A_i, B_{i+1}, $i \geq 0$, are $m \times m$ nonnegative matrices such that $\sum_{i=0}^{+\infty} A_i$ and $\sum_{i=1}^{+\infty} B_i$ are row stochastic.

These types of Markov chains, introduced by M.F. Neuts [23] model a very large variety of queueing problems. Due to the strong structure of P, it is possible to give an elegant representation of the probability invariant vector and of other performance measures of interest [23], in spite of the infinite size of the matrix.

A specific feature of M/G/1 type Markov chains is the block Hessenberg form of the matrix P and the block Toeplitz structure of the submatrix obtained by removing the first block row of P. Here we recall that a block matrix has a block Toeplitz structure if its blocks are constant along each block diagonal.

More general M/G/1 type Markov chains (non-skip-free) are defined by transition matrices of the form

$$P = \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} & \dots \\ \vdots & \vdots & \vdots & \\ B_{k,1} & B_{k,2} & B_{k,3} & \dots \\ A_0 & A_1 & A_2 & \dots \\ & A_0 & A_1 & \ddots \\ \bigcirc & & \ddots & \ddots \end{bmatrix} \qquad (2)$$

where A_i, $B_{j,i+1}$, $i \geq 0$, $j = 1, \dots, k$, are $m \times m$ nonnegative matrices such that $\sum_{i=0}^{+\infty} A_i$ and $\sum_{i=1}^{+\infty} B_{j,i}$, $j = 1, \dots, k$, are row stochastic.

The latter class of M/G/1 type Markov chains can be reduced to the block Hessenberg form (1) by partitioning the matrix P into blocks $\mathcal{A}_i$, $\mathcal{B}_i$ of dimension $p = mk$, where

$$\mathcal{A}_0 = \begin{bmatrix} A_0 & A_1 & \dots & A_{k-1} \\ 0 & A_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_1 \\ 0 & \dots & 0 & A_0 \end{bmatrix}, \mathcal{A}_i = \begin{bmatrix} A_{ik} & A_{ik+1} & \dots & A_{ik+k-1} \\ A_{ik-1} & A_{ik} & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{ik+1} \\ A_{ik-k+1} & \dots & A_{ik-1} & A_{ik} \end{bmatrix}, i \geq 1, \qquad (3)$$

$$\mathcal{B}_i = \begin{bmatrix} B_{1,(i-1)k+1} & B_{1,(i-1)k+2} & \dots & B_{1,ik} \\ B_{2,(i-1)k+1} & B_{2,(i-1)k+2} & \dots & B_{2,ik} \\ \vdots & \vdots & \vdots & \vdots \\ B_{k,(i-1)k+1} & B_{k,(i-1)k+2} & \dots & B_{k,ik} \end{bmatrix}, \; i \geq 1.$$

A fundamental step in solving Markov chains of M/G/1 type is computing the stochastic matrix G that solves the nonlinear matrix equation

$$X = \sum_{i=0}^{+\infty} A_i X^i \tag{4}$$

where X is an $m \times m$ matrix. The knowledge of G allows us to compute the probability invariant vector $\boldsymbol{\pi}^T = [\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \dots]$, partitioned according to the block structure of (1), by means of the recursive formula (Ramaswami's formula [25]):

$$\boldsymbol{\pi}_i = (I - \bar{A}_1^T)^{-1} \left[\bar{B}_i^T \boldsymbol{\pi}_0 + \sum_{j=1}^{i-1} \bar{A}_{i+1-j}^T \boldsymbol{\pi}_j \right], \; i \geq 1 \tag{5}$$

where

$$\bar{A}_i = \sum_{j=i}^{+\infty} A_j G^{j-i}, \quad \bar{B}_i = \sum_{j=i}^{+\infty} B_j G^{j-i}, \quad \text{for } i \geq 1; \tag{6}$$

the vector $\boldsymbol{\pi}_0$ is the solution of the system

$$\begin{cases} \bar{B}_1^T \boldsymbol{\pi}_0 = \boldsymbol{\pi}_0 \\ \left(\mathbf{e}^T + \mathbf{e}^T (I - \sum_{i=1}^{+\infty} \bar{A}_i^T)^{-1} \sum_{i=2}^{+\infty} \bar{B}_i^T \right) \boldsymbol{\pi}_0 = 1. \end{cases}$$

In many applications, the dimension m of the blocks A_i can be very large. For this reason it is important to design efficient algorithms for solving the nonlinear matrix equation (4), for computing Ramaswami's formula (5), and for solving the more general case of non-skip-free M/G/1 type Markov chains.

Recently, the analysis of numerical methods for solving the nonlinear matrix equation (4) has been developed by several authors. Classical linearly convergent methods based on functional iterations, proposed in [24], [16], have been studied in detail in [22], where an accurate analysis of the speed of convergence is provided. An adaptation of Newton's method is proposed in [17]. An efficient quadratically convergent algorithm for solving QBD problems is presented in [18]. A doubling algorithm that solves a sequence of nested block Toeplitz block Hessenberg systems is analyzed in [19]. A method based on the use of invariant subspaces is proposed in [1]. In the case of non-skip-free M/G/1 type Markov chains, an elegant analysis of the structure of the matrix G associated with the reblocked transition matrix (2), is provided in [13]; the authors observe that the matrix G is uniquely defined by a $k \times k$ block Frobenius matrix with blocks of size m, that is uniquely defined by the first block row $\boldsymbol{g}$ of G; in order to evaluate $\boldsymbol{g}$ an iterative technique, providing a sequence $\boldsymbol{g}_n$ of block vectors that monotonically converges to $\boldsymbol{g}$, is also proposed.

In this paper we show how the block Toeplitz structure of M/G/1 type Markov chains can be exploited as the basis for devising fast and numerically stable algorithms, for the computation of the probability invariant vector and for the solution of the nonlinear matrix equation (4), also in the general non-skip-free case. We show how the strong connection between polynomials and Toeplitz matrices allows us to devise a fast version of Ramaswami's formula, based on the computation of the inverse of an infinite block triangular block Toeplitz matrix, by means of Fast Fourier Transforms (FFT's). We present an improvement of the doubling algorithm of Latouche and Stewart [19] for solving (4), based on the concept of displacement rank: due to the Toeplitz structure of the problem, we show that only few block vectors need to be computed in order to solve the block Hessenberg block Toeplitz systems at each step of the algorithm, thus substantially reducing the computational cost. The classical cyclic reduction method is rephrased in functional form; this formulation allows us to devise a quadratically convergent and numerically stable algorithm, based on the use of FFT, for solving (4). An adaptation of this algorithm to QBD problems, leading to a method similar to that of Latouche and Ramaswami [18], is presented. Finally we consider the more general case of a non-skip-free M/G/1 type Markov chain; we present a new quadratically convergent algorithm based on cyclic reduction and on the concept of displacement rank; the properties of displacement rank allow us to fully exploit the specific Toeplitz structure of the blocks $\mathcal{A}_i$ of (3), thus dramatically reducing the computational cost of the cyclic reduction, while preserving the nice features of the quadratic convergence. Numerical results are presented.

The paper is organized as follows. In Section 2 we recall some basic tools for Toeplitz matrix computations. In Section 3 we present the fast version of Ramaswami's formula. In Section 4 we analyze the quadratically convergent algorithm for solving (4) based on the doubling technique of [19] and on cyclic reduction. In Section 5 we consider the case of non-skip-free M/G/1 type Markov chains. In Section 6 we present numerical results. In Appendix A, details on fast polynomial arithmetic are provided.

2 TOEPLITZ MATRIX TOOLS

In this section we recall the definition of Toeplitz matrices and the basic properties used for devising fast algorithms for computing the vector $\boldsymbol{\pi}$ and for computing the matrix G.

A block (possibly infinite) matrix A is a *block Toeplitz matrix* if its block entries are constant along each block diagonal.

2.1 TOEPLITZ MATRICES AND POLYNOMIALS

Matrix polynomials, matrix power series and block Toeplitz matrices are intimately related. The formal relation among the coefficients of the matrix power series $P(z) = \sum_{i=0}^{+\infty} P_i z^i$, $Q(z) = \sum_{i=0}^{+\infty} Q_i z^i$, and $R(z) = \sum_{i=0}^{+\infty} R_i z^i$, where $R(z) = P(z)Q(z)$ and P_i, Q_i, R_i, $i \geq 0$, are $m \times m$ matrices, can be easily rewritten in matrix form as

$$\begin{bmatrix} R_0 & & & O \\ R_1 & R_0 & & \\ R_2 & R_1 & R_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} P_0 & & & O \\ P_1 & P_0 & & \\ P_2 & P_1 & P_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} Q_0 & & & O \\ Q_1 & Q_0 & & \\ Q_2 & Q_1 & Q_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

In this way computations involving block triangular block Toeplitz matrices can be reduced to manipulating formal matrix power series (for infinite matrices) or polynomials (for finite matrices), and viceversa. Moreover, if $P(z)$ is a matrix polynomial of degree $n-1$ and $Q(z)$ is a matrix polynomial of degree $n/2-1$, then, by 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

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

Thus, the product between a finite block Toeplitz matrix and a block vector can be reduced to multiplying two matrix polynomials. Similarly, the inverse of a block triangular Toeplitz matrix is a block triangular Toeplitz matrix, uniquely defined by its first block column, that can be computed by inverting a matrix polynomial. In fact, 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 block lower triangular block Toeplitz matrix

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

In Appendix A we describe fast algorithms based on FFT for computing products and inverses of matrix polynomials. There, it is shown that due to this connection with matrix polynomial arithmetic, we may perform fast block Toeplitz matrix computation by using the technique of evaluation/interpolation at the roots of 1, based on FFT. More precisely, the computational cost needed to compute the product of an $n \times n$ block Toeplitz matrix and a block vector, or the product between two $n \times n$ block lower triangular block Toeplitz matrices, reduces to $O(m^3 n + m^2 n \log n)$ arithmetic operations; similarly, the same bound holds for the computational cost needed to compute the inverse of an $n \times n$ block Toeplitz matrix. Due to the use of FFT's, block Toeplitz matrix computation is weakly numerically stable [20]; we say that an algorithm is *weakly numerically stable* if the relative rounding errors in the output can be only bounded in norm.

2.2 DISPLACEMENT RANK

The concept of displacement rank, introduced by Kailath et al. in [15], is fundamental in devising and analyzing algorithms related to Toeplitz matrices [9], [8], [3].

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 $E_n^{(1)} = [I, O, \dots, O]^T$, $E_n^{(2)} = [O, \dots, O, I]^T$ 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 by the name n-dimensional *block column vector* any $mn \times m$ matrix; similarly we call by the name 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 $\Delta(H) = \sum_{i=1}^{r} U^{(i)} V^{(i)}$.

It can be easily observed that, if H is a block Toeplitz matrix, than H has block displacement rank 2. We call by the name *block Toeplitz-like matrix* an $n \times n$ block matrix having block displacement rank r which is independent of the block dimension n.

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

$$\Delta(H^{-1}) = -H^{-1}\Delta(H)H^{-1}. \tag{8}$$

From this property it follows in particular that the inverse of a block Toeplitz, or Toeplitz-like, matrix is a block Toeplitz-like matrix.

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

Theorem 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, and $L_n^T(W)$ is its transposed matrix.

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 (8) yielding the following result

$$K_n^{-1} = L_n(K_n^{-1} E_n^{(1)}) + \sum_{i=1}^{r} L_n(K_n^{-1} U^{(i)}) L_n^T(Z_n K_n^{-T} V^{(i)T}). \tag{9}$$

From the above results it follows that a block Toeplitz-like matrix, and its inverse, can be expressed by means of a sum of few products between block lower triangular and block upper triangular block Toeplitz matrices. This fact allows us to perform block Toeplitz-like matrix computations in terms of a few multiplications between matrix polynomials, in the light of the results of Section 2.1, with a substantial reduction of the computational cost.

3 FAST RAMASWAMI FORMULA

By means of the natural isomorphism between formal power series and triangular Toeplitz matrices outlined in Section 2.1, 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. We refer the reader to [21] for details.

We show that the block Hessenberg matrix

$$H = \begin{bmatrix} I - A_1^T & -A_0^T & & \bigcirc & \\ -A_2^T & I - A_1^T & -A_0^T & & \\ -A_3^T & -A_2^T & I - A_1^T & -A_0^T & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} \tag{10}$$

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

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

G is the solution of (4) and $\bar{A}_i$ are defined in (6). Ramaswami's formula (5) can now be derived by solving the system

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

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

$$L \begin{bmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \boldsymbol{\pi}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \bar{B}_1^T \\ \bar{B}_2^T \\ \bar{B}_3^T \\ \vdots \end{bmatrix} \boldsymbol{\pi}_0. \tag{11}$$

Hence, by solving the block lower triangular system (11) by forward substitution, we arrive at (5).

The computation of the vector $\boldsymbol{\pi}$ is thus reduced to solving a block lower triangular block Toeplitz infinite system. In [21] a fast Ramaswami formula, based on the use of FFT for solving the above infinite system, has been devised with a substantial reduction of the asymptotic computational cost. More precisely, the computational cost to calculate q block components of the probability invariant vector $\boldsymbol{\pi}$ is roughly $q(7m^2 + 40\log n_0)$ 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 $2qn_0m^2$ arithmetic operations. Due to the use of FFT's, the algorithm is weakly numerically stable.

4 COMPUTATION OF THE MATRIX G

In this section we present quadratically convergent methods for computing the matrix G. These methods consists of solving the infinite block Hessenberg block Toeplitz system

$$[X_1, X_2, X_3, \ldots]H = [A_0^T, O, O, \ldots] \tag{12}$$

where the matrix H is given in (10) and X_i are $m \times m$ matrices. In fact, it can be shown [12] that the solution of the above system is given by $X_1 = G^T$, $X_i = X_1^i$, $i \geq 2$.

4.1 A DOUBLING ALGORITHM

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

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

obtained by truncating the infinite matrix (10) and consider the finite system

$$[X_1^{(n)}, X_2^{(n)}, \ldots, X_n^{(n)}]H_n = [A_0^T, O, \ldots, O]. \tag{14}$$

In [19] the authors observe that the sequence $X_1^{(n)}$ monotonically converges, as n tends to infinity, to G^T. Hence, they propose a doubling algorithm for solving the finite system (14).

Let us assume for simplicity that $n = 2^q$, for a positive integer q. 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

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

and I denotes the $m \times m$ identity matrix. By applying the Sherman-Woodbury-Morrison formula [14] to the decomposition

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

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

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

By means of the above relations, in [19], [26], a doubling algorithm for the solution of the linear system (12), truncated at a finite system of block dimension n, has been devised, with a computational cost of $O(m^3 n \log^2 n)$ arithmetic operations. This algorithm can be much simplified by applying the tool of displacement rank of Section 2.2.

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

$$\Delta(H_n) = - \begin{bmatrix} I \\ O \\ \vdots \\ O \end{bmatrix} [A_0^T, O, \dots, O] + \begin{bmatrix} O \\ \vdots \\ O \\ I \end{bmatrix} [O, \dots, O, A_0^T]. \tag{15}$$

Using this, the equation (9) can be much simplified by noting that due to the special structure (15) of $\Delta(H_n)$ we may easily prove that

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

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 explicitly 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 the use of FFT. In this way we arrive at a weakly numerically stable 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 relations among block rows and columns of H_n^{-1} and $H_{\frac{n}{2}}^{-1}$.

Theorem 2 *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*

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

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

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

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

where $P = (I - A_0^T R_{\frac{n}{2}}^{(1)} T_{\frac{n}{2}} C_{\frac{n}{2}}^{(2)})^{-1} A_0^T$.

4.2 CYCLIC REDUCTION BASED ALGORITHM

In this section we propose a method for the numerical solution of (12), introduced in [5], [7], [6], which is quadratically convergent, weakly numerically stable and, unlike the method of Section 4.1, avoids the need of cutting the infinite system (12) 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 fast algorithms based on the use of FFT.

Our method relies on the technique of successive state reduction (cyclic reduction [10]), applied to the infinite system (12).

By performing an odd-even permutation of the block rows and block columns in (12) we find that

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

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

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

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 (12).

In other words, $H^{(1)}$ is uniquely 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 (12). Each matrix $Q^{(j)}$ is 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 $A^{(j)}(z) = \sum_{i=0}^{+\infty} A_i^{(j)} z^i$, $\widehat{A}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{A}_{i+1}^{(j)} z^i$, respectively, where $A_i^{(0)} = \widehat{A}_i^{(0)} = A_i^T$, $i \geq 1$, $A_0^{(0)} = A_0^T$. Then the following functional relations hold:

$$\begin{cases} A^{(j+1)}(z) = zA_{odd}^{(j)}(z) + A_{even}^{(j)}(z)\left(I - A_{odd}^{(j)}(z)\right)^{-1} A_{even}^{(j)}(z) \\ \widehat{A}^{(j+1)}(z) = \widehat{A}_{odd}^{(j)}(z) + A_{even}^{(j)}(z)\left(I - A_{odd}^{(j)}(z)\right)^{-1} \widehat{A}_{even}^{(j)}(z) \end{cases} \tag{18}$$

where

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

The relations (18), besides expressing 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 FFT's.

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

Theorem 3 *The blocks* $\{A_i^{(j)}\}_{i\geq 0}$, $\{\widehat{A}_i^{(j)}\}_{i\geq 1}$ *are nonnegative matrices such that* $\sum_{i=0}^{+\infty} A_i^{(j)}$, $A_0^T + \sum_{i=1}^{+\infty} \widehat{A}_i^{(j)}$ *are column stochastic. Moreover, if the matrix* $I - \sum_{i=0}^{+\infty} G^{T^{i\cdot 2^j}} \widehat{A}_{i+1}^{(j)}$ *is nonsingular, then* $G^T = A_0^T \left(I - \sum_{i=0}^{+\infty} G^{T^{i\cdot 2^j}} \widehat{A}_{i+1}^{(j)}\right)^{-1}$.

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 [6], [7], [5]:

Theorem 4 *Let* $G' = \lim_{j\to\infty} G^{T^j}$. *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 (4) is irreducible then* $\lim_{j\to\infty} A_0^{(j)}(I - A_1^{(j)})^{-1} = G'$ *and* $\lim_{j\to\infty} A_i^{(j)} = 0$, $i \geq 2$.

The use of FFT techniques allows us to reduce the computational cost of the j-th step of the cyclic reduction algorithm to $O(m^3 n_j + m^2 n_j \log n_j)$ arithmetic operations, where n_j is such that the matrices $A_i^{(j)}$ are negligible for $i > n_j$.

In the case of QBD problems, where $A_i = O$ for $i > 2$, functional relations (18) are simplified, leading to an algorithm similar to that of Latouche and Ramaswami [18], but with a lower computational cost:

$$\begin{aligned}
A_0^{(j+1)} &= A_0^{(j)}(I - A_1^{(j)})^{-1} A_0^{(j)} \\
A_1^{(j+1)} &= A_1^{(j)} + A_0^{(j)}(I - A_1^{(j)})^{-1} A_2^{(j)} + A_2^{(j)}(I - A_1^{(j)})^{-1} A_0^{(j)} \\
A_2^{(j+1)} &= A_2^{(j)}(I - A_1^{(j)})^{-1} A_2^{(j)} \\
\widehat{A}_1^{(j+1)} &= \widehat{A}_1^{(j)} + A_0^{(j)}(I - A_1^{(j)})^{-1} \widehat{A}_2^{(j)} \\
\widehat{A}_2^{(j+1)} &= A_2^{(j+1)}.
\end{aligned}$$

For details on the implementation of the algorithm we refer the reader to papers [7] and [5], where two different FFT based techniques are proposed for the computation of the coefficients of the matrix power series $A^{(j)}(z)$ and $\widehat{A}^{(j)}(z)$. A Fortran 90 code implementing the version described in [5] can be obtained by anonymous ftp at `morse.dm.unipi.it` in the directory /pub/MARKOV.

5 SOLVING NON-SKIP-FREE M/G/1 TYPE MARKOV CHAINS

The properties of displacement rank and the nice features of the cyclic reduction of Section 4.2 allow us to devise a new efficient quadratically convergent algorithm for computing the matrix G associated with reblocked non-skip-free M/G/1 type Markov chains.

Let us introduce the following notations: the block vectors $E_1 = [I, O, \ldots, O]^T$, $E_k = [O, \ldots, O, I]^T$ denote $mk \times m$ matrices made up by the first and the last m columns of the identity matrix of order mk.

Consider the matrices $\mathcal{A}_i$ of (3) obtained by reblocking the matrix (2) and consider the sequence of matrix power series $\mathcal{A}^{(j)}(z)$, $\widehat{\mathcal{A}}^{(j)}(z)$ obtained by means of functional relations (18), where each A is replaced with $\mathcal{A}$. We may easily observe that the formal matrix power series $\mathcal{H}^{(0)}(z) = \sum_{i=0}^{+\infty} \mathcal{H}_i^{(0)} z^i = zI - \mathcal{A}^{(0)}(z)$ and $\widehat{\mathcal{H}}^{(0)}(z) = \sum_{i=0}^{+\infty} \widehat{\mathcal{H}}_i^{(0)} z^i = I - \widehat{\mathcal{A}}^{(0)}(z)$ are block Toeplitz matrices. At a first glance, the Toeplitz structure looks lost by the matrix power series $\mathcal{H}^{(j)}(z) = \sum_{i=0}^{+\infty} \mathcal{H}_i^{(j)} z^i = zI - \mathcal{A}^{(j)}(z)$ and $\widehat{\mathcal{H}}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{\mathcal{H}}_i^{(j)} z^i I - \widehat{\mathcal{A}}^{(j)}(z)$, for $j \geq 1$. However, the block Toeplitz-like structure is kept at each step j of cyclic reduction, as stated by the following:

Theorem 5 *For the matrix power series* $\mathcal{H}^{(j)}(z) = zI - \mathcal{A}^{(j)}(z)$, $\widehat{\mathcal{H}}^{(j)}(z) = I - \widehat{\mathcal{A}}^{(j)}(z)$ *generated at* j*-th step of cyclic reduction we have*

$$\begin{array}{l} \Delta(\mathcal{H}^{(j)}(z)) = (\mathcal{H}^{(j)}(z)E_1)U^{(j)}(z) - V^{(j)}(z)(E_k^T\mathcal{H}^{(j)}(z)) \\ \Delta(\widehat{\mathcal{H}}^{(j)}(z)) = -E_1(E_k^T\mathcal{A}_0^{(0)}) - (\mathcal{H}^{(j)}(z)E_1)\widehat{U}^{(j)}(z) - V^{(j)}(z)(E_k^T\widehat{\mathcal{H}}^{(j)}(z)) \end{array} \tag{19}$$

where $U^{(j)}(z) = \sum_{i=0}^{+\infty} U_i^{(j)} z^i$, $\widehat{U}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{U}_i^{(j)} z^i$ *are block row vector power series,* $V^{(j)}(z) = \sum_{i=0}^{+\infty} V_i^{(j)} z^i$ *is a block column vector power series, defined by*

$$\begin{array}{rcl} U^{(j)}(z^2) & = & -U^{(j-1)}(z)(\mathcal{H}^{(j-1)}(z) - \mathcal{H}^{(j-1)}(-z))^{-1}\mathcal{H}^{(j-1)}(-z) + \\ & & U^{(j-1)}(-z)(\mathcal{H}^{(j-1)}(z) - \mathcal{H}^{(j-1)}(-z))^{-1}\mathcal{H}^{(j-1)}(z) \\ \widehat{U}^{(j)}(z^2) & = & (\widehat{U}^{(j-1)}(z) - \widehat{U}^{(j-1)}(-z))/(2z) + (U^{(j-1)}(z) - U^{(j-1)}(-z)) \\ & & (\mathcal{H}^{(j-1)}(z) - \mathcal{H}^{(j-1)}(-z))^{-1}(\widehat{\mathcal{H}}^{(j-1)}(z) - \widehat{\mathcal{H}}^{(j-1)}(-z))/(2z), \\ V^{(j)}(z^2) & = & -\mathcal{H}^{(j-1)}(-z)(\mathcal{H}^{(j-1)}(z) - \mathcal{H}^{(j-1)}(-z))^{-1}V^{(j-1)}(z) + \\ & & \mathcal{H}^{(j-1)}(z)(\mathcal{H}^{(j-1)}(z) - \mathcal{H}^{(j-1)}(-z))^{-1}V^{(j-1)}(-z) \end{array}$$

for $j \geq 1$*, and* $U^{(0)}(z) = zE_k^T$, $\widehat{U}^{(0)}(z) = E_k^T$, $V^{(0)}(z) = zE_1$.

Proof. The proof follows by induction on j. □

From the above theorem, the following representations hold for the matrix power series $\mathcal{H}^{(j)}(z)$ and $\widehat{\mathcal{H}}^{(j)}(z)$:

Theorem 6 *At each step* j *of cyclic reduction it holds that*

$$\begin{array}{rcl} \mathcal{H}^{(j)}(z) & = & L(\mathcal{H}^{(j)}(z)E_1)(I - L^T(Z_kU^{(j)}(z)^T)) + L(V^{(j)}(z))L^T(Z_k\mathcal{H}^{(j)}(z)^TE_k) \\ \widehat{\mathcal{H}}^{(j)}(z) & = & L(\widehat{\mathcal{H}}^{(j)}(z)E_1) + L^T(Z_k\mathcal{A}_0^{(0)T}E_k) + \\ & & L(\mathcal{H}^{(j)}(z)E_1)L^T(Z_k\widehat{U}^{(j)}(z)^T) + L(V^{(j)}(z))L^T(Z_k\widehat{\mathcal{H}}^{(j)}(z)^TE_k). \end{array}$$

Proof. The proof simply follows from Theorem 5 and from the results of Section 2.2. □

From the above results it follows that, at each step j, only the seven block vectors power series $U^{(j)}(z)$, $\widehat{U}^{(j)}(z)$, $V^{(j)}(z)$, $\mathcal{C}^{(j)}(z) = \mathcal{H}^{(j)}(z)E_1$, $\mathcal{R}^{(j)}(z) = E_k^T\mathcal{H}^{(j)}(z)$, $\widehat{\mathcal{C}}^{(j)}(z) = \widehat{\mathcal{H}}^{(j)}(z)E_1$, $\widehat{\mathcal{R}}^{(j)}(z) = E_k^T\widehat{\mathcal{H}}^{(j)}(z)$, need to be computed in order to represent the matrix power series $\mathcal{A}^{(j)}(z)$ and $\widehat{\mathcal{A}}^{(j)}(z)$.

Explicit relations between the block vector power series $\mathcal{C}^{(j)}(z) = \sum_{i=0}^{+\infty} \mathcal{C}_i^{(j)} z^i$, $\mathcal{R}^{(j)}(z) = \sum_{i=0}^{+\infty} \mathcal{R}_i^{(j)} z^i$, $\widehat{\mathcal{C}}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{\mathcal{C}}_i^{(j)} z^i$, $\widehat{\mathcal{R}}^{(j)}(z) = \sum_{i=0}^{+\infty} \widehat{\mathcal{R}}_i^{(j)} z^i$ at two subsequent steps of cyclic reduction can be obtained by multiplying functional relations (18) on the left by E_k^T, and on the right by E_1.

Moreover the block vectors defining $\mathcal{A}^{(j)}(z)$ and $\widehat{\mathcal{A}}^{(j)}(z)$ are related to the block vectors obtained at step $j-1$, which define $\mathcal{A}^{(j-1)}(z)$ and $\widehat{\mathcal{A}}^{(j-1)}(z)$, by means of functional relations that only involve sums of products between block triangular block Toeplitz matrices. Hence, the computational cost needed to perform j-th step of cyclic reduction by using FFT's reduces to $O(n_j m^3 k \log^2 k + m^2 k n_j \log n_j)$, where n_j is such that $\mathcal{A}_i^{(j)}$ and $\widehat{\mathcal{A}}_i^{(j)}$ are negligible for $i > n_j$. We refer the reader to [4] for details.

In the case where $\mathcal{A}_i = O$ for $i > 2$ the above results are further simplified. We can give explicit expressions of the blocks $\mathcal{H}_0^{(j)} = -\mathcal{A}_0^{(j)}$, $\mathcal{H}_1^{(j)} = I - \mathcal{A}_1^{(j)}$, $\mathcal{H}_2^{(j)} = -\mathcal{A}_2^{(j)}$, $\widehat{\mathcal{H}}_1^{(j)} = I - \widehat{\mathcal{A}}_1^{(j)}$ by means of the following theorems:

Theorem 7 *If $\mathcal{A}_i = O$ for $i > 2$, then for the matrices $\mathcal{H}_0^{(j)}$, $\mathcal{H}_1^{(j)}$, $\mathcal{H}_2^{(j)}$, $\widehat{\mathcal{H}}_1^{(j)}$, $j \geq 0$, generated by the cyclic reduction (18) we have*

$$
\begin{aligned}
\Delta(\mathcal{H}_0^{(j)}) &= \mathcal{C}_0^{(j)} U_0^{(j)} - V_0^{(j)} \mathcal{R}_0^{(j)} \\
\Delta(\mathcal{H}_1^{(j)}) &= \mathcal{C}_0^{(j)} U_1^{(j)} + \mathcal{C}_1^{(j)} U_0^{(j)} - V_0^{(j)} \mathcal{R}_1^{(j)} - V_1^{(j)} \mathcal{R}_0^{(j)} \\
\Delta(\mathcal{H}_2^{(j)}) &= \mathcal{S}^{(j)} U_1^{(j)} - V_1^{(j)} \widehat{\mathcal{R}}_1^{(j)} \\
\Delta(\widehat{\mathcal{H}}_1^{(j)}) &= -E_1 \mathcal{R}_0^{(0)} - \mathcal{C}_0^{(j)} U_1^{(j)} - V_0^{(j)} \widehat{\mathcal{R}}_1^{(j)},
\end{aligned}
$$

where $\mathcal{S}^{(0)} = \mathcal{C}_1^{(0)}$, $\mathcal{S}^{(j)} = \mathcal{S}^{(j-1)} - \mathcal{H}_2^{(j-1)} \mathcal{H}_1^{(j-1)^{-1}} \mathcal{C}_0^{(j-1)}$, for $j \geq 1$.

Proof. The proof follows by comparing the terms of the same degree in formula (19). □

Theorem 8 *At each step j of cyclic reduction, the matrices $\mathcal{H}_0^{(j)}$, $\mathcal{H}_1^{(j)}$, $\mathcal{H}_2^{(j)}$, $\mathcal{H}_1^{(j)^{-1}}$, $\widehat{\mathcal{H}}_1^{(j)}$ can be rewritten as*

$$
\begin{aligned}
H_0^{(j)} = \quad & L(\mathcal{C}_0^{(j)})(I - L^T(Z_k U_0^{(j)T})) + L(V_0^{(j)}) L^T(Z_k \mathcal{R}_0^{(j)T}) \\
H_1^{(j)} = \quad & L(\mathcal{C}_1^{(j)})(I - L^T(Z_k U_0^{(j)T})) - L(\mathcal{C}_0^{(j)}) L^T(Z_k U_1^{(j)T}) + L(V_0^{(j)}) L^T(Z_k \mathcal{R}_1^{(j)T}) + \\
& L(V_1^{(j)}) L^T(Z_k \mathcal{R}_0^{(j)T}) \\
H_2^{(j)} = \quad & L(\mathcal{C}_2^{(j)}) - L(\mathcal{S}^{(j)}) L^T(Z_k U_1^{(j)T}) + L(V_1^{(j)}) L^T(Z_k \widehat{\mathcal{R}}_1^{(j)T}) \\
H_1^{(j)^{-1}} = \quad & L(\mathcal{H}_1^{(j)^{-1}} E_1) + L^T(Z_k \mathcal{H}_1^{(j)^{-T}} U_0^{(j)T}) + L(\mathcal{H}_1^{(j)^{-1}} \mathcal{C}_0^{(j)}) L^T(Z_k \mathcal{H}_1^{(j)^{-T}} U_1^{(j)T}) - \\
& L(\mathcal{H}_1^{(j)^{-1}} V_1^{(j)}) L^T(Z_k \mathcal{H}_1^{(j)^{-T}} \mathcal{R}_0^{(j)T}) \\
\widehat{H}_1^{(j)} = \quad & L(\widehat{\mathcal{C}}_1^{(j)}) + L^T(Z_k \mathcal{R}_0^{(0)T}) - L(\mathcal{C}_0^{(j)}) L^T(Z_k U_1^{(j)T}) + L(V_0^{(j)}) L^T(Z_k \widehat{\mathcal{R}}_1^{(j)T}).
\end{aligned}
$$

Proof. The proof follows from Theorem 7 and by the results of Section 2.2. □

Explicit relations for the vectors $U_i^{(j)}$, $V_i^{(j)}$ $\mathcal{C}_i^{(j)}$, $\mathcal{R}_i^{(j)}$, $\widehat{\mathcal{C}}_i^{(j)}$, $\widehat{\mathcal{R}}_i^{(j)}$, can be provided:

$$
\begin{aligned}
U_0^{(j)} &= U_0^{(j-1)} - U_1^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_0^{(j-1)}\\
U_1^{(j)} &= -U_1^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_2^{(j-1)}\\
V_0^{(j)} &= V_0^{(j-1)} - \mathcal{H}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}V_1^{(j-1)}\\
V_1^{(j)} &= -\mathcal{H}_2^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}V_1^{(j-1)}\\
\mathcal{C}_0^{(j)} &= -\mathcal{H}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{C}_0^{(j-1)}\\
\mathcal{C}_1^{(j)} &= \mathcal{C}_1^{(j-1)} - \mathcal{H}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{C}_2^{(j-1)} - \mathcal{H}_2^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{C}_0^{(j-1)}\\
\mathcal{C}_2^{(j)} &= -\mathcal{H}_2^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{C}_2^{(j-1)}\\
\mathcal{R}_0^{(j)} &= -\mathcal{R}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_0^{(j-1)}\\
\mathcal{R}_1^{(j)} &= \mathcal{R}_1^{(j-1)} - \mathcal{R}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_2^{(j-1)} - \mathcal{R}_2^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_0^{(j-1)}\\
\mathcal{R}_2^{(j)} &= -\mathcal{R}_2^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_2^{(j-1)}\\
\widehat{\mathcal{R}}_1^{(j)} &= \widehat{\mathcal{R}}_1^{(j-1)} - \mathcal{H}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{R}_2^{(j-1)}\\
\widehat{\mathcal{C}}_1^{(j)} &= \widehat{\mathcal{C}}_1^{(j-1)} - \mathcal{C}_0^{(j-1)}\mathcal{H}_1^{(j-1)^{-1}}\mathcal{H}_2^{(j-1)}
\end{aligned}
$$

for $j \geq 1$, where $U_0^{(0)} = V_0^{(0)} = 0$, $U_1^{(0)} = E_k^T$, $V_0^{(0)} = E_1$. For this particular case, the computational cost per step is reduced to $O(m^3 k \log^2 k)$ operations.

For the blocks $\mathcal{A}_i^{(j)}$, $\widehat{\mathcal{A}}_i^{(j)}$ the quadratically convergence properties of the blocks $A_i^{(j)}$, $\widehat{A}_i^{(j)}$ of Section 4.2 still hold. Hence, due to the displacement structure of the blocks $\mathcal{A}_i$ of (3), we have devised a quadratically convergent algorithm, with low computational cost, for non-skip-free M/G/1 type Markov chains. This new algorithm particularly shows its effectiveness, with respect to the linearly convergent method of [13], when the block dimension k is large; in fact, the computational cost of one step of the latter method is $O(mk^2)$ and many iterations may need to be computed in order to reach a good approximation of G.

Numerical experiments which show the efficiency of our approach are presented in next section.

6 NUMERICAL RESULTS

We have tested our algorithms for computing the matrix G on a problem arising from the mathematical modeling of a Metaring MAC Protocol [2]. For this particular example the blocks A_i have dimension $m = 16$. 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 [23].

The integer n_0 such that the blocks A_i are negligible for $i > n_0$ 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 cyclic reduction algorithm with the functional iteration method based on the recursion

$$\begin{aligned} &X_0 = I, \\ &X_{n+1} = A_0 \left(I - \sum_{i=2}^{+\infty} A_i X_n^{i-1}\right)^{-1}, \; n \geq 0, \end{aligned} \tag{20}$$

which is the fastest among the classical linearly convergent functional iterations [22].

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), the number of iterations needed by the Cyclic Reduction (CR) algorithm to compute an approximation of the matrix G; the residual of the computed approximation is also reported. We compared cyclic reduction

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: Cyclic Reduction

with classical techniques based on functional iterations: Table 2 reports the CPU time (in seconds) and the number of iterations needed by the Functional Iteration Formula (FIF) defined in (20), the residual of the computed approximation, and the ratio between the times needed by this algorithm and by CR. Finally we have considered the problem corresponding to $\rho = 0.8$ and we have compared the CPU time needed by the doubling algorithm based on displacement rank of Section 4.1 and the Latouche-Stewart algorithm [19]. Table 3 reports the number of CPU seconds required by our algorithm (algorithm DR: Displacement Rank), together with the residual error, for the values of the block dimension n of (14) ranging from 256 to 4096. It reports also the time required by the Latouche–Stewart algorithm [19] (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.

ρ	Time (s.)	Iterations	Residual	FIF/CR
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 2: Functional Iteration Method

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 3: Fast and customary doubling algorithm

We have also tested the cyclic reduction based algorithm for computing the matrix G associated with reblocked non-skip-free M/G/1 type Markov chains. We have randomly generated blocks A_i of dimension $m = 4$, such that $A_i = 0$ for $i > 2k$, for different values of k. That is, we have considered the simple case where $A_i = 0$ for $i > 2$. We have compared the above algorithm (NSF-CR Algorithm) with cyclic reduction algorithm for QBD problem of Section 4.2 (CR Algorithm), and with the algorithm [13] of Gail, Hantler, Taylor (GHT Algorithm). Table 4 reports the CPU time (in seconds) and the number of iterations needed by NSF-CR, CR and GHT algorithms to compute an approximation of the matrix G; the residual of the computed approximation is also reported. The symbol "*" denotes that the algorithm is not able to run for lack of memory.

	NSF-CR			CR			GHT		
k	Steps	Time	Residual	Steps	Time	Residual	Steps	Time	Residual
16	10	14.4	$5.9 \cdot 10^{-16}$	10	1.1	$9.9 \cdot 10^{-16}$	1534	6.1	$9.9 \cdot 10^{-11}$
32	9	19.2	$6.3 \cdot 10^{-15}$	9	9.8	$1.4 \cdot 10^{-15}$	848	13.3	$9.9 \cdot 10^{-11}$
64	9	37.7	$5.4 \cdot 10^{-13}$	9	136.3	$2.4 \cdot 10^{-14}$	980	60.4	$9.9 \cdot 10^{-11}$
128	10	89.5	$1.2 \cdot 10^{-11}$	*	*	*	2229	555.5	$9.9 \cdot 10^{-11}$
256	9	146.4	$1.9 \cdot 10^{-15}$	*	*	*	800	816.4	$9.9 \cdot 10^{-11}$
512	12	448.1	$3.1 \cdot 10^{-11}$	*	*	*	1380	5830.3	$1.7 \cdot 10^{-4}$

Table 4: Non-skip-free Markov chains

A MATRIX POLYNOMIAL ARITHMETIC

Throughout this section $\mathbf{i}$ is the imaginary unit such that $\mathbf{i}^2 = -1$, $\omega = \cos\frac{2\pi}{n} + \mathbf{i}\sin\frac{2\pi}{n}$ denotes a principal n-th root of 1, $\bar{x}$ denotes the complex conjugate of $x \in \boldsymbol{C}$. The vector $\boldsymbol{y} = (y_j) \in \boldsymbol{C}^n$ such that $y_j = \sum_{i=0}^{n-1} x_i \omega^{ij}$ is called the discrete Fourier transform of the vector $\boldsymbol{x} = (x_j) \in \boldsymbol{C}^n$, and is denoted by $\boldsymbol{y} = \mathrm{DFT}_n(\boldsymbol{x})$; the vector $\boldsymbol{x} = \mathrm{IDFT}_n(\boldsymbol{y})$, $\boldsymbol{x} = (x_j)$, $x_j = \frac{1}{n}\sum_{i=0}^{n-1} y_i \bar{\omega}^{ij}$ denotes the inverse discrete Fourier transform of $\boldsymbol{y}$.

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 [11].

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 $U_s = P(\omega^s)$, $V_s = Q(\omega^s)$, $s = 0, \ldots, n-1$, $\omega = \cos\frac{2\pi}{n} + \boldsymbol{i} \sin\frac{2\pi}{n}$, in the following way

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

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.

Consider now the problem of inverting a polynomial, i.e., of solving the congruence $P(z)Q(z) = I \bmod z^n$.

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

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

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

$$T_n^{-1} = \begin{pmatrix} T_{n/2}^{-1} & O \\ -T_{n/2}^{-1} H_{n/2} T_{n/2}^{-1} & T_{n/2}^{-1} \end{pmatrix} \tag{21}$$

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 (21) with $n = 2^{i+1}$ in the following way: by applying Algorithm 1 the products $W = H_{2^i} U$ and $V = T_{2^i}^{-1} W$.

Observe that at the i-th stage of Algorithm 2 the computation of 3 DFT's and 2 IDFT's of order 2^{i+1} must be performed, together with the multiplication of about 2^{i+1} complex $m \times m$ matrices. The overall cost is $O(m^3 n + m^2 n \log n)$ ops.

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*, 1997. To appear.

[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. Effective methods for solving banded Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 1997. Submitted for publication.

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

[6] 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.

[7] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17(4):906–926, 1996.

[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] D.F. Elliott and K.R. Rao. *Fast Transform Algorithms, Analyses, Applications*. Academic Press, New York, 1982.

[12] 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.

[13] 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.

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

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

[16] 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.

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

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

[19] 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.

[20] E. Linzer. On the stability of transform-based circular deconvolution. *SIAM J. Num. Anal.*, 29:1482–1492, 1992.

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

[22] 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.

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

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

[25] 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.

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