# An improved FFT-based version of Ramaswami's formula

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

**Abstract**

Ramaswami's formula allows a numerically stable computation of the probability invariant vector of a stochastic matrix of $M/G/1$ type. We observe that this formula consists in solving, by means of forward substitution, a block lower triangular block Toeplitz infinite system. By using the natural isomorphism between matrix polynomials and block Toeplitz matrices, we derive a new fast algorithm, based on the use of block FFT's, for the computation of Ramaswami's formula. The proposed algorithm is particularly effective when many block components of the probability invariant vector must be computed.

**Key words.** $M/G/1$ type matrices, Ramaswami's formula, Toeplitz matrices, Fast Fourier Transform.

# 1 INTRODUCTION

In this paper, in order to agree with the customary notation of polynomial computation theory, we adopt transposed notations with respect to those standard in Markov chain theory.

A stochastic matrix of $M/G/1$ type [14] is an infinite row stochastic matrix $P^T$ of the form

$$P = \begin{pmatrix} B_0 & A_0 & & \bigcirc & \\ B_1 & A_1 & A_0 & & \\ B_2 & A_2 & A_1 & A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix}, \tag{1}$$

where $B_i$, $A_i$, $i \geq 0$, are $k \times k$ matrices. Observe that, except for the first block column, $P$ is a block Toeplitz matrix, i.e., a matrix with block entries that depend only on the difference of their indices. Suppose that $P$ is irreducible and positive recurrent [7] and denote by $\boldsymbol{\pi}$ its invariant probability vector, that is, the solution to the equation

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

where here and hereafter $\mathbf{e}$ denotes the vector, of suitable dimension, having all the components equal to 1. According to the structure of $P$, we partition the vector $\boldsymbol{\pi}$ into the vectors $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, .... The block components $\boldsymbol{\pi}_1$, $\boldsymbol{\pi}_2$, $\boldsymbol{\pi}_3$, ... can be computed by directly solving the above system (for example by means of the methods proposed in [2], [17]), or by applying a numerically stable recursive formula, once the solution of a suitable nonlinear matrix equation has been calculated. More precisely, let $G$ be the column stochastic matrix given by the minimal nonnegative solution of the matrix equation

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

and define the matrices

$$\bar{A}_{i+1} = \sum_{j=i+1}^{+\infty} G^{j-i-1} A_j, \quad \bar{B}_i = \sum_{j=i}^{+\infty} G^{j-i} B_j, \quad \text{for } i \geq 0. \tag{4}$$

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

$$\bar{B}_0 \boldsymbol{\pi}_0 = \boldsymbol{\pi}_0, \quad \mathbf{k}^T \boldsymbol{\pi}_0 = 1,$$

where

$$\mathbf{k}^T = \mathbf{e}^T + \mathbf{e}^T(I - \sum_{i=1}^{+\infty} \bar{A}_i)^{-1} \sum_{i=1}^{+\infty} \bar{B}_i,$$

the vectors $\boldsymbol{\pi}_i$, for $i \geq 1$, are given by *Ramaswami's formula* [15]

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

As there exist many efficient numerical methods to solve the matrix equation (3) (see [3], [8], [10], [13]) and since the matrix $G$ is useful in calculating performance measures [14], Ramaswami's formula is an effective tool for the computation of the probability invariant vector associated with M/G/1 type Markov chains.

Formula (5) was first proved by Ramaswami in [15], by using results from Markov renewal theory (Ramaswami's proof may also be found in [14]). In [16], formula (5) was proved by using a different technique, based on the computation of the Schur complement of a leading principal submatrix of $I - P$. In both the approaches, the proof consists in building, by suitably truncating the matrix $P$ of (1), the sequence $\{P^{(n)}\}_n$ of column stochastic $(n+1) \times (n+1)$ block matrices, whose probability invariant vector is given, up to a multiplicative constant, by the corresponding components of the infinite vector $\boldsymbol{\pi}$. Specifically, in [15] and [16], by using different tools, the authors prove the following result, which underlies Ramaswami's formula.

**Theorem 1.1** *Let $n \in \mathbf{N}$ and consider the matrix*

$$P^{(n)} = \begin{pmatrix} B_0 & A_0 & & & \\ B_1 & A_1 & A_0 & & \bigcirc \\ \vdots & \vdots & \ddots & \ddots & \\ B_{n-1} & A_{n-1} & \dots & A_1 & A_0 \\ \bar{B}_n & \bar{A}_n & \dots & \bar{A}_2 & \bar{A}_1 \end{pmatrix},$$

*where $\bar{A}_i$ and $\bar{B}_i$ are defined in (4). Then,*
*– the matrix $P^{(n)}$ is an irreducible and positive recurrent column stochastic matrix;*
*– the probability invariant vector $\mathbf{y}^{(n)}$ associated with $P^{(n)}$, made up by the block components $\mathbf{y}_0^{(n)}$, $\mathbf{y}_1^{(n)}$, …, $\mathbf{y}_n^{(n)}$, satisfies $\mathbf{y}_i^{(n)} = \gamma_n \boldsymbol{\pi}_i$, $0 \leq i \leq n$, for a suitable constant $\gamma_n > 0$.*

By using generating functions associated with triangular Toeplitz matrices, we give a novel interpretation of Ramaswami's formula and a new fast algorithm, based on the use of FFT techniques, for its computation. More precisely, by means of the natural isomorphism between formal power series and triangular Toeplitz matrices, we find a block $UL$ factorization ($U$ is a block upper triangular matrix and $L$ is a block lower triangular matrix) of the block Hessenberg matrix

$$H = \begin{pmatrix} I-A_1 & -A_0 & & \bigcirc & \\ -A_2 & I-A_1 & -A_0 & & \\ -A_3 & -A_2 & I-A_1 & -A_0 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{6}$$

Then we derive the formula of Ramaswami by solving the system $(I-P)\boldsymbol{\pi} = \mathbf{0}$ by means of that block $UL$ factorization. The computation of the vector $\boldsymbol{\pi}$ is so reduced to solving a block lower triangular block Toeplitz infinite system. If that system is solved by means of forward substitution we obtain formula (5). By using this technique, the computational cost to calculate $p$ block components of the probability invariant vector $\boldsymbol{\pi}$ grows in the variable $p$ as $2pm_0k^2$ arithmetic operations, where $m_0$ is an integer such that for $i > m_0$ the matrices $A_i$ are negligible. If the system is solved by FFT-based computations, we obtain a new algorithm, with a substantial reduction of the asymptotic computational cost. More precisely, the computational cost to calculate $p$ block components of the probability invariant vector $\boldsymbol{\pi}$ grows in $p$ at most as $p(7k^2 + 40\log m_0)$ arithmetic operations. We observe that the advantages are particularly strong in the case where the blocks $A_i$ are negligible only for large values of $i$ and in the case where many block components of the vector $\boldsymbol{\pi}$ need to be computed.

With regards to numerical stability, the customary Ramaswami formula is strongly numerically stable, since all the computations require only additions of nonnegative numbers, multiplications and divisions. The FFT-based version is weakly numerically stable [11], i.e., we are able to give upper bound only to the norm of the relative errors generated in the computations. However, in many numerical experiments performed, we detected no appreciable differences between the results computed with the two algorithms.

The paper is organized as follows. In Section 2 we derive the Ramaswami formula by means of the block $UL$ factorization of the matrix $H$. In Section 3 we introduce the basic tools for FFT's computations and we propose the new fast algorithm for the computation of the vector $\boldsymbol{\pi}$. In Section 4 we present some numerical results.

# 2 RAMASWAMI'S FORMULA AND BLOCK $UL$ FACTORIZATION

Let $A$ be an infinite block lower triangular block Toeplitz matrix, defined by the $k \times k$ matrices $\alpha_0, \alpha_1, \alpha_2, \ldots$, i.e.

$$A = \begin{pmatrix} \alpha_0 & & & & \\ \alpha_1 & \alpha_0 & & \bigcirc & \\ \alpha_2 & \alpha_1 & \alpha_0 & & \\ \alpha_3 & \alpha_2 & \alpha_1 & \alpha_0 & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

We associate the matrix $A$ with the formal $k \times k$ matrix power series $\varphi_A(z) = \sum_{i=0}^{+\infty} \alpha_i z^i$, called the *generating function* for the matrix $A$. Generating functions allow us to represent row by column multiplication and inversion of block lower triangular block Toeplitz matrices. In fact, the following results can be easily shown

**Proposition 2.1** *Let $A$ and $B$ be two infinite block lower triangular block Toeplitz matrices, with blocks of the same dimension. Let $\varphi_A(z)$ and $\varphi_B(z)$, respectively, be their matrix-generating functions. Then the matrix $C = AB$ is a block lower triangular block Toeplitz matrix whose matrix-generating function $\varphi_C(z)$ satisfies $\varphi_C(z) = \varphi_A(z)\varphi_B(z)$, the matrix product of $\varphi_A(z)$ and $\varphi_B(z)$.*

**Corollary 2.2** *Let $A$ be an infinite block lower triangular block Toeplitz matrix with matrix-generating function $\varphi_A(z)$. If $A$ is nonsingular, i.e., $\det \alpha_0 \neq 0$, then the inverse $A^{-1}$ is a block lower triangular block Toeplitz matrix and $\varphi_{A^{-1}}(z)$ satisfies $\varphi_A(z)\varphi_{A^{-1}}(z) = I$.*

Consider the block Toeplitz block lower triangular matrix, defined by the second block column of the matrix $I - P$, where $P$ is given by (1), and associate with it the matrix generating function $\varphi(z) = zI - \sum_{i=0}^{+\infty} A_i z^i$. The following result holds:

**Proposition 2.3** *Let $G$ be the minimal nonnegative solution of the matrix equation (3) and let $\bar{A}_i$, for $i \geq 1$, be defined in (4). Then*

$$\varphi(z) = \Big(zI - G\Big)\Big(I - \sum_{i=0}^{+\infty} \bar{A}_{i+1} z^i\Big). \tag{7}$$

**Proof.** Immediate, by comparing the coefficients of $z^i$ in both sides of (7), since $A_i = \bar{A}_i - G\bar{A}_{i+1}$, for $i \geq 1$. □

For Proposition 2.1, the functional decomposition given in Proposition 2.3 can be rephrased in terms of block lower triangular block Toeplitz matrices factorization.

**Corollary 2.4** *The matrix $H$ of (6) can be factorized as the product $H = UL$, where*

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

**Proof.** The proof readily follows by taking the matrix equation corresponding to decomposition (7) and by discarding the first block row. □

Ramaswami's formula can now be derived by solving the system

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

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

$$L\begin{pmatrix}\boldsymbol{\pi}_1\\ \boldsymbol{\pi}_2\\ \boldsymbol{\pi}_3\\ \vdots\end{pmatrix}=\begin{pmatrix}\bar{B}_1\\ \bar{B}_2\\ \bar{B}_3\\ \vdots\end{pmatrix}\boldsymbol{\pi}_0. \tag{8}$$

Hence, by solving the block lower triangular system (8) by forward substitution, we arrive at (5).

# 3 RAMASWAMI'S FORMULA COMPUTATION BY MEANS OF FFT-BASED TECHNIQUES

The block lower triangular block Toeplitz system (8) can be solved by using FFT-based techniques, thus obtaining a fast and weakly numerically stable method for the computation of the probability invariant vector of (2). Before explaining the method, we introduce some basic tools for Toeplitz matrices computations by means of FFT's.

## 3.1 TOEPLITZ MATRICES COMPUTATION

Matrix polynomials computation and block Toeplitz matrices computation are closely related. Specifically,

- the product of an $(m/2)\times(m/2)$ block Toeplitz matrix and a block vector can be performed by means of a matrix polynomial multiplication modulo $z^m-1$;
- the inversion of an $m\times m$ block lower triangular block Toeplitz matrix can be performed by inverting a matrix polynomial modulo $z^m$.

We will therefore present fast algorithms for matrix polynomial computations, by extending to the block case the FFT-based algorithms for scalar polynomials (see [4], [3]).

Throughout this section we will denote $\mathbf{y}=(y_0,\ldots,y_{d-1})=\mathrm{DFT}(\mathbf{x})$ the Discrete Fourier Transform of the vector $\mathbf{x}=(x_0,\ldots,x_{d-1})$ defined by $y_i=$

$\sum_{j=0}^{d-1} \omega^{ij} x_j$, $i = 0, \ldots, d-1$, 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 $x_i = \frac{1}{d}\sum_{j=0}^{d-1} \omega^{-ij} y_j$, $i = 0, \ldots, d-1$. It is well known that if $d = 2^M$, where $M$ is a positive integer, the computation of DFT and IDFT can be performed with $O(d \log d)$ real arithmetic operations (hereafter denoted by ops) by means of the base-2 FFT algorithms [4]. More precisely, in the case of real input the cost of DFT and IDFT is $\frac{5}{2} d \log d + O(1)$ ops and $\frac{5}{2} d \log d + d + O(1)$ ops, respectively, if we do not count the cost of computing the $d$-th roots of 1.

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

Consider the problem of the matrix polynomial multiplication modulo $z^m - 1$. Let $P_i$, $i = 0, \ldots, m-1$, be $k \times l$ matrices, and let $Q_i$, $i = 1, \ldots, m-1$, be $l \times h$ matrices. Let $P(z) = \sum_{i=0}^{m-1} P_i z^i$, $Q(z) = \sum_{i=0}^{m-1} Q_i z^i$, $R(z) = \sum_{i=0}^{m-1} R_i z^i$ be matrix polynomials such that $R(z) = P(z)Q(z) \bmod (z^m - 1)$. Then the $k \times h$ matrix coefficients of $R(z)$ can be computed, given the matrix coefficients of $P(z)$ and $Q(z)$, by means of the following algorithm.

**Algorithm 3.1** *Computation of matrix polynomial product modulo $z^m - 1$.*

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

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

**Computation.**

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

$$(U_0, \ldots, U_{m-1}) = \mathrm{DFT}(P_0, \ldots, P_{m-1}),$$

$$(V_0, \ldots, V_{m-1}) = \mathrm{DFT}(Q_0, \ldots, Q_{m-1}).$$

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

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

The cost of Algorithm 3.1 is $O\big(klhm + (kl + lh + kh)m \log m\big)$ ops. More precisely, for real input matrices the cost is less than $3(kl + lh + kh)m \log m + m(3klh + 2kh)$. Indeed, due to the real input we have $W_s = \bar{W}_{m-s}$, $s = 1, \ldots, m/2 - 1$, where $\bar{W}_{m-s}$ is the complex conjugate of $W_{m-s}$, moreover $W_0$, $W_{m/2}$ are real matrices. Thus the computation at stage 2 is reduced to $m/2 - 1$ complex matrix products and 2 real matrix products. Complex matrix products are computed by means of the algorithm which uses 3 multiplications and 5 additions [6].

Observe that Algorithm 3.1 can be also used to compute efficiently the product of a block Toeplitz matrix and a vector. In fact, if $P(z)$ has degree $m-1$ and $Q(z)$ has degree $m/2 - 1$, then by comparing the terms of degree $m/2, \ldots, m-1$ in the equation $R(z) = P(z)Q(z) \bmod (z^m - 1)$, we get the following equation involving a block Toeplitz matrix

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

Thus, that product can be computed by means of Algorithm 3.1, i.e., by evaluating a product of two matrix polynomials. The polynomials $P(z)$ and $Q(z)$ have coefficients $P_0, P_1, \ldots, P_{m-1}$, $Q_0, Q_1, \ldots, Q_{m-1}$, where $P_0 = Q_{m/2} = \ldots = Q_{m-1} = O$.

Consider now the inversion of a polynomial modulo $z^m$. Suppose that $P_i$ and $Q_i$, $i = 0, \ldots, m-1$, are square matrices of order $k$. Comparing the coefficients of $z^i$ in both sides of the equation $P(z)Q(z) = I \bmod z^m$ yields the equation involving a lower block triangular block Toeplitz matrix

$$\begin{pmatrix} P_0 & & & \bigcirc \\ P_1 & P_0 & & \\ \vdots & \ddots & \ddots & \\ P_{m-1} & \ldots & P_1 & P_0 \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{m-1} \end{pmatrix} = \begin{pmatrix} I \\ O \\ \vdots \\ O \end{pmatrix}. \tag{9}$$

Observe that, since the inverse of a lower block triangular block Toeplitz matrix is also lower block triangular block Toeplitz, the block vector $(Q_0^T, Q_1^T, \ldots, Q_{m-1}^T)^T$ determines the inverse of the matrix in (9).

Denote by $T_m$ the matrix in (9), assume for simplicity that $m = 2^M$, $M$ positive integer, and partition $T_m$ as

$$T_m = \begin{pmatrix} T_{m/2} & O \\ H_{m/2} & T_{m/2} \end{pmatrix}$$

where all blocks are $(m/2) \times (m/2)$ block Toeplitz matrices. Then $T_m^{-1}$ can be written as

$$T_m^{-1} = \begin{pmatrix} T_{m/2}^{-1} & O \\ -T_{m/2}^{-1} H_{m/2} T_{m/2}^{-1} & T_{m/2}^{-1} \end{pmatrix} \quad (10)$$

and the first block column $(Q_0^T, \ldots, Q_{m-1}^T)^T$ of $T_m^{-1}$, which solves (9), can be computed from the first block column $(Q_0^T, \ldots, Q_{m/2-1}^T)^T$ of $T_{m/2}^{-1}$ by means of two multiplications of a $(m/2) \times (m/2)$ block Toeplitz matrix by a block vector. So, we arrive at the following algorithm which extends the algorithm of Lafon and Sieveking-Kung to the block case [5].

**Algorithm 3.2** *Solution of the congruence $P(z)Q(z) = I \bmod z^m$, or, equivalently, of the system (9).*

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

**Output.** The $k \times k$ matrices $Q_0, Q_1, \ldots, Q_{m-1}$ satisfying (9), or equivalently, such that the polynomial $Q(z) = \sum_{i=0}^{m-1} Q_i z^i$ solves the congruence $P(z)Q(z) = I \bmod z^m$.

**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 (10) with $m = 2^{i+1}$ in the following way: by applying Algorithm 3.1 compute the products $W = H_{2^i} U$ and $V = T_{2^i}^{-1} W$.

Observe that, at the $i$-th stage of Algorithm 3.2, 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 $k \times k$ matrices. The overall cost is about $6mk^3 + \frac{25}{2} k^2 m \log m$ ops.

## 3.2 COMPUTATION OF RAMASWAMI'S FORMULA

Observe that, since the matrices $\sum_{i=0}^{+\infty} A_i$ and $\sum_{i=0}^{+\infty} B_i$ are stochastic, the matrices $A_i$ and $B_i$ are negligible for large values of $i$. Therefore, the blocks $A_i$ and $B_i$ can be set equal to zero for indices $i > m_0$, where $m_0$ is such that $\sum_{i=0}^{m_0} A_i$ and $\sum_{i=0}^{m_0} B_i$ are numerically stochastic (a nonnegative matrix $A$ is said numerically stochastic if $\mathbf{e}^T(I - A) < \mu \mathbf{e}^T$, where $\mu$ is the machine precision). The matrices $\bar{A}_i$ and $\bar{B}_i$ are then also equal to zero for $i > m_0$, and (8) reduces to a block banded system with bandwidth $m_0$. Solving the banded system (8) by forward substitution leads to the Ramaswami algorithm:

**Algorithm 3.3** *CRF – Customary Ramaswami's Formula*

**Input.** Integers $p$ and $m_0$, the matrix $G$ solution of (3) and the matrices $A_i$, $B_i$, $i=0,\dots,m_0$, such that $\sum_{i=0}^{m_0} A_i$, $\sum_{i=0}^{m_0} B_i$ are numerically stochastic.

**Output.** Components $\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_p$ of the probability invariant vector $\boldsymbol{\pi}$ solving (2) and the value $\sigma(p) = \sum_{i=0}^{p} \mathbf{e}^T \boldsymbol{\pi}_i$.

**Computation.**

1. Computation of matrices $\bar{A}_i$, $i=0,\dots,m_0$, $\bar{B}_i$, $i=1,\dots,m_0$:
   (a) Set $\bar{A}_{m_0} = A_{m_0}$, $\bar{B}_{m_0} = B_{m_0}$.
   (b) For $i = m_0-1,\dots,1$ compute $\bar{A}_i = A_i + G\bar{A}_{i-1}$ and $\bar{B}_i = B_i + G\bar{B}_{i-1}$.
   (c) Compute $\bar{B}_0 = B_0 + G\bar{B}_1$.
2. Computation of $\boldsymbol{\pi}_0$:
   (a) Solve the system $\bar{B}_0\boldsymbol{\pi}_0 = \boldsymbol{\pi}_0$, $\mathbf{k}^T\boldsymbol{\pi}_0 = 1$, where $\mathbf{k}^T = \mathbf{e}^T + \mathbf{e}^T(I - \sum_{i=1}^{m_0} \bar{A}_i)^{-1}(\sum_{i=1}^{m_0} \bar{B}_i)$.
   (b) Set $\sigma(p) = \mathbf{e}^T\boldsymbol{\pi}_0$.
3. Computation of $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_p$:
   (a) Compute $\boldsymbol{\pi}_1 = (I - \bar{A}_1)^{-1}\bar{B}_1\boldsymbol{\pi}_0$;
   (b) For $i = 2,\dots,p$, compute

$$\begin{aligned}\boldsymbol{\pi}_i &= (I-\bar{A}_1)^{-1}\left[\bar{B}_i\boldsymbol{\pi}_0 + \textstyle\sum_{j=j_0}^{i-1} \bar{A}_{i+1-j}\boldsymbol{\pi}_j\right]\\ \sigma(p) &= \sigma(p) + \mathbf{e}^T\boldsymbol{\pi}_i\end{aligned}$$

   where $j_0 = \min(1, i+1-m_0)$ and $\bar{B}_i = O$ for $i > m_0$.

The computational cost of step 1 and 2 is about $4m_0k^3$ ops; the computational cost of step 3, if $p > m_0$, is about $k^3 + k^2m_0^2 + 2k^2m_0(p-m_0)$ ops. Therefore, the asymptotic computational cost to calculate $p$ components of $\boldsymbol{\pi}$, where $p > m_0$, by means of Algorithm 3.3 is $2pk^2m_0$. That computational cost can be substantially reduced by solving system (8) by the FFT-based algorithms for Toeplitz computation described in Section 3.1. Suppose, without loss of generality, that $m_0 = 2^M$ (if $m_0$ is not a power of 2, we truncate the sequences $A_i$, $B_i$ at the least power of 2 greater than $m_0$). Let us partition the matrix $L$ of (8) into blocks of dimension $km_0$, where each block is itself partitioned into $m_0$ sub-blocks of dimension $k$. So, the matrix $L$ can be rewritten as

$$L = \begin{pmatrix} I-R & & & \bigcirc \\ -S & I-R & & \\ & -S & I-R & \\ \bigcirc & & \ddots & \ddots \end{pmatrix}, \tag{11}$$

where

$$R = \begin{pmatrix} \bar{A}_1 & & & & \bigcirc \\ \bar{A}_2 & \bar{A}_1 & & & \\ \bar{A}_3 & \bar{A}_2 & \bar{A}_1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \bar{A}_{m_0} & \bar{A}_{m_0-1} & \dots & \bar{A}_2 & \bar{A}_1 \end{pmatrix}, S = \begin{pmatrix} O & \bar{A}_{m_0} & \dots & \bar{A}_3 & \bar{A}_2 \\ & O & \bar{A}_{m_0} & \dots & \bar{A}_3 \\ & & \ddots & \ddots & \vdots \\ & & & O & \bar{A}_{m_0} \\ \bigcirc & & & & O \end{pmatrix}$$

are block triangular block Toeplitz matrices. Let us partition the vector $\boldsymbol{\pi}$ according to the structure (11) of $L$, i.e., $\boldsymbol{\pi}^T = (\tilde{\boldsymbol{\pi}}_0^T, \tilde{\boldsymbol{\pi}}_1^T, \tilde{\boldsymbol{\pi}}_2^T, \dots)^T$, where $\tilde{\boldsymbol{\pi}}_0 = \boldsymbol{\pi}_0$, $\tilde{\boldsymbol{\pi}}_i^T = (\boldsymbol{\pi}_{m_0(i-1)+1}^T, \boldsymbol{\pi}_{m_0(i-1)+2}^T, \dots, \boldsymbol{\pi}_{m_0(i-1)+m_0}^T)^T$. With these notations the system (8) can be solved by means of the equations

$$\begin{aligned} \tilde{\boldsymbol{\pi}}_1 &= (I-R)^{-1}\mathbf{b} \\ \tilde{\boldsymbol{\pi}}_i &= (I-R)^{-1}S\tilde{\boldsymbol{\pi}}_{i-1}, \quad i \geq 2, \end{aligned} \tag{12}$$

where

$$\mathbf{b} = \begin{pmatrix} \bar{B}_1 \\ \bar{B}_2 \\ \vdots \\ \bar{B}_{m_0} \end{pmatrix} \boldsymbol{\pi}_0. \tag{13}$$

The matrices involved in (12) are block triangular block Toeplitz matrices. The computations can be performed by the FFT-based methods of Section 3.1, leading to the following fast algorithm for $\boldsymbol{\pi}$:

**Algorithm 3.4** *FRF – Fast Ramaswami's Formula*

**Input.** Integers $n$ and $m_0$, the matrix $G$ solution of (3) and the matrices $A_i$, $B_i$, $i = 0, \dots, m_0$, such that $\sum_{i=0}^{m_0} A_i$, $\sum_{i=0}^{m_0} B_i$ are numerically stochastic.

**Output.** Components $\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_p$ of the probability invariant vector $\boldsymbol{\pi}$, solving (2), where $p = nm_0$, and the value $\sigma(p) = \sum_{i=0}^{p} \mathbf{e}^T \boldsymbol{\pi}_i$.

**Computation.**

1. Computation of matrices $\bar{A}_i$, $i = 0, \dots, m_0$, $\bar{B}_i$, $i = 1, \dots, m_0$:
   (a) Set $\bar{A}_{m_0} = A_{m_0}$, $\bar{B}_{m_0} = B_{m_0}$.
   (b) For $i = m_0 - 1, \dots, 1$ compute $\bar{A}_i = A_i + G\bar{A}_{i-1}$, $\bar{B}_i = B_i + G\bar{B}_{i-1}$.
   (c) Compute $\bar{B}_0 = B_0 + G\bar{B}_1$.
2. Computation of $\boldsymbol{\pi}_0$:
   (a) Solve the system $\bar{B}_0 \boldsymbol{\pi}_0 = \boldsymbol{\pi}_0$, $\mathbf{k}^T \boldsymbol{\pi}_0 = 1$, where $\mathbf{k}^T = \mathbf{e}^T + \mathbf{e}^T(I - \sum_{i=1}^{m_0} \bar{A}_i)^{-1}(\sum_{i=1}^{m_0} \bar{B}_i)$.
   (b) Set $\sigma(p) = \mathbf{e}^T \boldsymbol{\pi}_0$.

3. Computation of $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_p$:
   (a) Compute $(I-R)^{-1}$ by Algorithm 3.2.
   (b) Compute, by Algorithm 3.1, $\tilde{\boldsymbol{\pi}}_1 = (I-R)^{-1}\mathbf{b}$, where $\mathbf{b}$ is given in (13), and compute $\sigma(p) = \sigma(p) + \mathbf{e}^T\tilde{\boldsymbol{\pi}}_1$.
   (c) For $i = 2, \ldots, n$, compute, by Algorithm 3.1,

$$\mathbf{s} = S\tilde{\boldsymbol{\pi}}_{i-1}, \quad \tilde{\boldsymbol{\pi}}_i = (I-R)^{-1}\mathbf{s},$$

   and $\sigma(p) = \sigma(p) + \mathbf{e}^T\tilde{\boldsymbol{\pi}}_i$.

Steps 1 and 2 in Algorithm 3.4 are the same as Steps 1 and 2 in Algorithm 3.3, and have the computational cost of $4m_0k^3$ ops. At Step 3, the computational cost of calculating stage (a) and (b) is about $6m_0k^3 + 35/2k^2m_0 \log m_0$; at stage (c), once the DFT of the matrix $S$ is computed (with less than $10k^2m_0 \log m_0$ ops.), the computational cost of the generic block $\tilde{\boldsymbol{\pi}}_i$ is at most $7k^2m_0 + 20km_0 \log m_0 + 20km_0 \leq 7k^2m_0 + 40km_0 \log m_0$ ops. Whence the overall cost to compute $p = nm_0$ components is at most $10m_0k^3 + 39m_0 \log m_0 + n(7k^2m_0 + 40km_0 \log m_0)$, i.e., the computational cost grows linearly in $p$ at most as $p(7k^2 + 40k \log m_0)$. Therefore Algorithm 3.4 is asymptotically (as $p$ tends to infinity) $2k^2m_0/(7k^2 + 40 \log m_0)$ times faster than Algorithm 3.3. Moreover Algorithm 3.4 is particularly faster in cases where $m_0$ is large, that is, in cases where the blocks $A_i$ and $B_i$ are negligible only for large values of $i$.

Concerning numerical stability, all computations in Algorithm 3.3 consist in performing additions of nonnegative numbers, multiplications and divisions, so that Algorithm 3.3 is strongly numerically stable, i.e., the relative errors generated by the floating point arithmetic can be bounded entry-wise. Algorithm 3.4, due to the use of FFT's, is weakly numerically stable, i.e., it is only possible to give very good relative error bounds in terms of the norm (see [11], [4]). However, in spite of the weak stability of FFT, in numerous experiments we have performed, we detected no difference between the results of the algorithms.

# 4 NUMERICAL RESULTS AND COMPARISONS

We implemented Algorithms 3.3 and 3.4 in Fortran 77; the programs were run on an Alpha Workstation using standard double precision IEEE arithmetic. We tested the algorithms on a BMAP/G/1 queue (Example 1) and on a problem arising from modeling a Metaring MAC Protocol [1] (Example 2).

**Example 1** We consider an 8-state BMAP/G/1 queue with constant service times. The blocks $A_i$ and $B_i$ have dimension 8 and have the structure described in [12], that is,

$$A_i^T = \sum_{j=0}^{+\infty} e^{-\theta} \frac{\theta^j}{j!} K_i^{(j)}, \quad B_i^T = -D_0^{-1} \sum_{j=0}^{i} D_{j+1} A_{i-j}^T,$$

where $\theta = max_h\{-(D_0)_{hh}\}$ and the $\{K_i^{(j)}\}$ are recursively defined by $K_0^{(0)} = I$, $K_i^{(0)} = O$, $i \geq 1$, and

$$K_0^{(j+1)} = K_0^{(j)}(I + \theta^{-1} D_0),$$

$$K_i^{(j+1)} = \theta^{-1} \textstyle\sum_{h=0}^{i-1} K_h^{(j)} D_{i-h} + K_i^{(j)}(I + \theta^{-1} D_0), \quad i \geq 1.$$

In our particular example,

$$\begin{array}{l} (D_i)_{jh} = e^{-\lambda_{jh}} \dfrac{\lambda_{jh}^i}{i!}, \ i \geq 1, \\ (D_0)_{jh} = e^{-\lambda_{jh}}, \ j \neq h, \\ (D_0)_{hh} = -\sum_{j \neq h} (D_0)_{jh} - \sum_{i=1}^{+\infty} \sum_{j=1}^{8} (D_i)_{jh}, \end{array}$$

for suitable nonnegative parameters $\lambda_{jh}$. The integer $m_0$ for which the matrices $\sum_{i=0}^{m_0} A_i$, $\sum_{i=0}^{m_0} B_i$ are numerically stochastic is 64. Table 1 reports, for different numbers $p$ of computed block components, the corresponding value $\sigma(p) = \sum_{i=0}^{p} \mathbf{e}^T \boldsymbol{\pi}_i$, the CPU times (in seconds) needed by Algorithm 3.3 and Algorithm 3.4, respectively, and their ratios. Figure 1 shows the CPU time needed by

| $p$ | $\sigma(p)$ | $CRF(sec.)$ | $FRF(sec.)$ | $CRF/FRF$ |
|---|---|---|---|---|
| $2^6$ | 0.077379 | 0.01 | 0.07 | 0.1 |
| $2^7$ | 0.148238 | 0.03 | 0.08 | 0.4 |
| $2^8$ | 0.274049 | 0.05 | 0.09 | 0.6 |
| $2^9$ | 0.472667 | 0.11 | 0.12 | 0.9 |
| $2^{10}$ | 0.721746 | 0.22 | 0.17 | 1.3 |
| $2^{11}$ | 0.922526 | 0.41 | 0.27 | 1.5 |
| $2^{12}$ | 0.993994 | 0.84 | 0.59 | 1.4 |
| $2^{13}$ | 0.999963 | 1.77 | 1.03 | 1.7 |
| $2^{14}$ | 0.999999 | 3.73 | 1.74 | 2.1 |

Table 1: BMAP/G/1 Queue

Algorithm 3.4 and Algorithm 3.3 to compute the same number of block components of the vector $\boldsymbol{\pi}$.

**Example 2** We consider a problem arising from modeling a metropolitan queueing problem (Metaring MAC Protocol [1]). The dimension $k$ of the blocks $A_i$ and

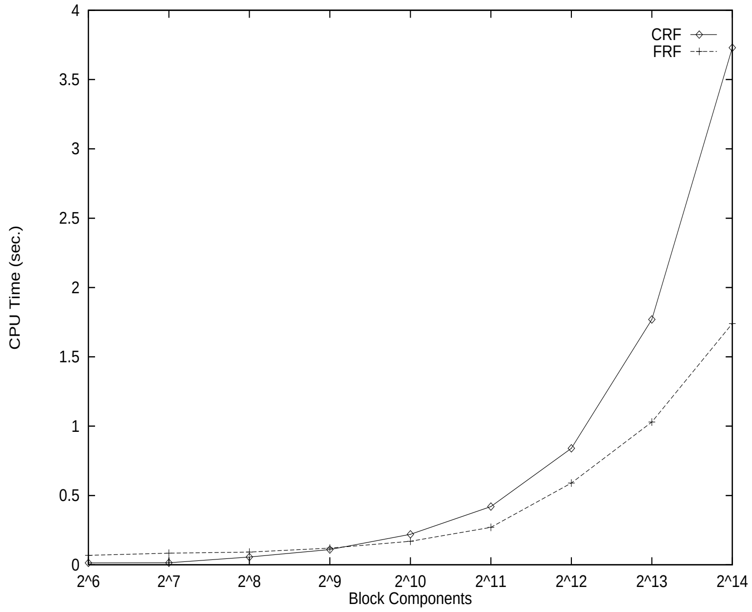


Figure 1: BMAP/G/1 Queue

$B_i$ is 80, and the integer $m_0$ such that the matrices $\sum_{i=0}^{m_0} A_i$, $\sum_{i=0}^{m_0} B_i$ are numerically stochastic is 256. For a description of the model and the structure of the blocks we refer to [1]. This particular example is characterized by large block bandwidth of the matrix $L$ in system (8), by a large dimension of the blocks $A_i$ and $B_i$, and by a long queue, i.e., the blocks $\boldsymbol{\pi}_i$ tend to zero slowly. Table 2 reports, for different numbers $p$ of block components, the corresponding value $\sigma(p) = \sum_{i=0}^{p} \mathbf{e}^T \boldsymbol{\pi}_i$, the CPU times (in seconds) needed by Algorithm 3.3 and Algorithm 3.4, respectively, and their ratios. Figure 2 shows the CPU time needed by Algorithm 3.4 and Algorithm 3.3 to compute the same number of block components of the vector $\boldsymbol{\pi}$.

Observe that, in both the examples, the customary Ramaswami formula is faster than Fast Ramaswami formula in the case where only few block components of the vector $\boldsymbol{\pi}$ need to be computed. Fast Ramaswami formula really shows its effectiveness when many block components must be computed. When the block dimension $k$ is large and the number $m_0$ of nonnegligible blocks $A_i$ and $B_i$ is large, the advantages of the FRF method are clear.

# References

[1] Anastasi G., Lenzini L., Meini B., "Performance evaluation of a worst

| $p$ | $\sigma(p)$ | $CRF(sec.)$ | $FRF(sec.)$ | $CRF/FRF$ |
|---|---|---|---|---|
| $2^9$ | 0.285677 | 181 | 374 | 0.5 |
| $2^{10}$ | 0.457895 | 385 | 399 | 1.0 |
| $2^{11}$ | 0.687780 | 797 | 424 | 1.9 |
| $2^{12}$ | 0.896434 | 1681 | 454 | 3.7 |
| $2^{13}$ | 0.988604 | 3236 | 487 | 6.6 |
| $2^{14}$ | 0.999862 | 6632 | 652 | 10.2 |
| $2^{15}$ | 0.999999 | 13322 | 829 | 16.1 |

Table 2: Metaring MAC Protocol

case model of the Metaring MAC Protocol with global fairness", *Performance Evaluation*, forthcoming, 1997.

[2] Bini D., Meini B. "On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems", *Computations with Markov Chains*, W. J. Stewart Ed., pp. 21–38, Kluwer Academic Publishers, 1996.

[3] Bini D., Meini B., "On the solution of a nonlinear matrix equation arising in queueing problems", *SIAM J. Matrix Anal. Appl.*, 17 (1996), to appear.

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

[5] Bini D., Pan V., "Polynomial division and its computational complexity", *J. Complexity*, 6 (1986), pp. 179–203.

[6] Borodin A., Munro I., *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York, 1975.

[7] Çinlar E., *Introduction to Stochastic Processes,* Prentice-Hall, Englewood Cliffs, N.J., 1975.

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

[9] Latouche G., Ramaswami V., "A logarithmic reduction algorithm for quasi-birth-death processes", *J. Appl. Prob.*, 30 (1993), pp. 650–674.

[10] Latouche G., Stewart G. W., "Numerical methods for M/G/1 type queues", *Computations with Markov Chains*, W. J. Stewart Ed., pp. 571–581, Kluwer Academic Publishers, 1996.

[11] Linzer E., "On the stability of transform-based circular deconvolution", *SIAM J. Numer. Anal.*, 29 (1992), pp. 1482–1492.

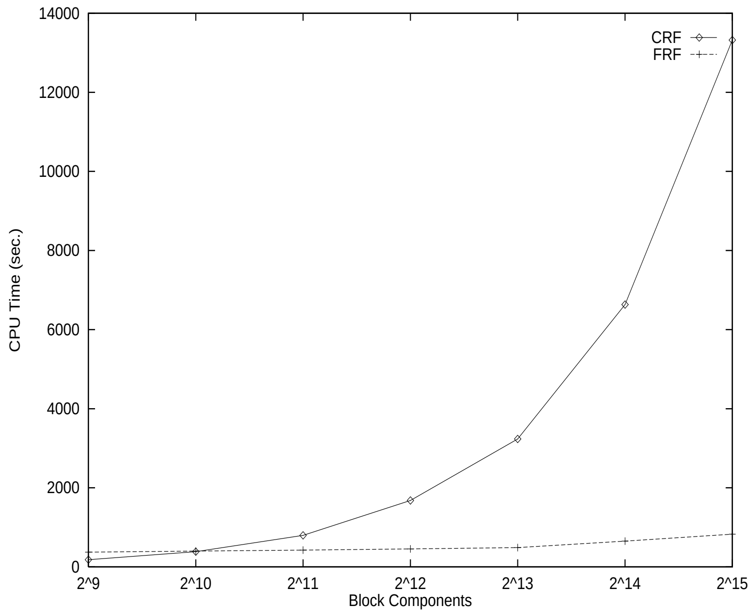


Figure 2: Metaring MAC Protocol


[12] Lucantoni D. M., "New results on the single server queue with a batch Markovian arrival process", *Commun. Statist. Stochastic Models,* 7 (1991), pp. 1–46.

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

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

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

[16] Schellhaas H., "On Ramaswami's algorithm for the computation of the steady state vector in Markov chains of $M/G/1$ type", *Commun. Statist. Stochastic Models,* 6 (1990), pp. 541–550.

[17] Stewart G. W., "On the solution of block Hessenberg systems", *Numer. Lin. Algebra with Appl.*, 2 (1995), pp. 287–296.