MAMSolver: A matrix analytic methods tool[1]

Alma Riska Evgenia Smirni
Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795
{riska,esmirni}@cs.wm.edu

1 Introduction

Over the last two decades, considerable effort has been put into the development of *matrix analytic techniques* [8, 9] for the exact analysis of a general and frequently encountered class of queuing models. In these models, the embedded Markov chains are two-dimensional generalizations of elementary GI/M/1 and M/G/1 queues and their intersection, i.e., quasi-birth-death (QBD) processes. Matrix analytic techniques have been successfully used for modeling of complex computer systems [14]. Representative classes of Markov chains that can be solved using matrix analytic techniques are the BMAP/PH/1 and MAP/PH/1 queues which can be used for modeling queuing systems that experience both correlation and high variability, distinct characteristics of Internet traffic. MAMSolver is a collection of the most efficient solution methodologies for M/G/1, GI/M/1 and QBD processes. In distinction to existing tools such as MAGIC [13] and MGMTool [4] which provide solutions only for QBD processes, MAMSolver provides implementations of the classical and the most efficient recent algorithms that solve M/G/1, GI/M/1 and QBD processes. MAMSolver computes the stationary probability vector for the process under study, simple measures of interest such as the average system queue length and response time, and probabilistic indicators such as the caudal characteristic [5].

2 MAMSolver framework

MAMSolver deals with the solution of M/G/1, GI/M/1, and QBD processes. Throughout this exposition, we refer to continuous time Markov processes. However MAMSolver provides solution for both continuous and discrete time Markov processes. The formal representation of the processes under study is given by the block partitioned infinitesimal generator:

$$\mathbf{Q}_{M/G/1} = \begin{bmatrix} \widehat{\mathbf{L}} & \widehat{\mathbf{F}}^{(1)} & \widehat{\mathbf{F}}^{(2)} & \widehat{\mathbf{F}}^{(3)} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L} & \mathbf{F}^{(1)} & \mathbf{F}^{(2)} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F}^{(1)} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \mathbf{Q}_{GI/M/1} = \begin{bmatrix} \widehat{\mathbf{L}} & \widehat{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}}^{(1)} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \mathbf{F} & \cdots \\ \widehat{\mathbf{B}}^{(3)} & \mathbf{B}^{(2)} & \mathbf{B}^{(1)} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \mathbf{Q}_{QBD} = \begin{bmatrix} \widehat{\mathbf{L}} & \widehat{\mathbf{F}} & \mathbf{0} & \mathbf{0} & \cdots \\ \widehat{\mathbf{B}} & \mathbf{L} & \mathbf{F} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B} & \mathbf{L} & \mathbf{F} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B} & \mathbf{L} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Block partitioning the infinitesimal generator defines a partition of the infinite state space in an infinite number of sets $\mathcal{S}^{(i)}$ for $i \geq 0$, and the corresponding stationary probability vector in sub-vectors $\boldsymbol{\pi}^{(i)}$ for $i \geq 0$.

2.1 M/G/1-type Processes

For the class of M/G/1-type continuous Markov chains, the infinitesimal generator is of the form $\mathbf{Q}_{M/G/1}$. Key to all algorithms that solve M/G/1-type processes is the computation of the matrix $\mathbf{G}$, which is given by the solution of the matrix equation:

$$\mathbf{B} + \mathbf{L}\mathbf{G} + \sum_{j=1}^{\infty} \mathbf{F}^{(j)} \mathbf{G}^{j+1} = \mathbf{0}. \quad (1)$$

[1]This work was partially supported by the National Science Foundation under grants EIA-9974992, CCR-0098278, and ACI-0090221.

Matrix $\mathbf{G}$ has important probabilistic interpretation: an entry (k,l) in $\mathbf{G}$ expresses the conditional probability of the process first entering $\mathcal{S}^{(j-1)}$ through state l, given that it starts from state k of $\mathcal{S}^{(j)}$ [9, page 81]. Recent advances show that the computation of $\mathbf{G}$ is more efficient when displacement structures are used based on the representation of M/G/1-type processes by means of QBD processes [1, 5]. The most efficient algorithm for the computation of $\mathbf{G}$ is cyclic-reduction [1]. The calculation of the stationary probability vector is based on Ramaswami's recursive formula [10], which defines the following recursive relation among stationary probability vectors $\boldsymbol{\pi}^{(i)}$ for $i \geq 0$:

$$\boldsymbol{\pi}^{(j)} = -\left(\boldsymbol{\pi}^{(0)}\widehat{\mathbf{S}}^{(j)} + \sum_{k=1}^{j-1}\boldsymbol{\pi}^{(k)}\mathbf{S}^{(j-k)}\right)\mathbf{S}^{(0)^{-1}} \quad \forall j \geq 1, \tag{2}$$

where $\widehat{\mathbf{S}}^{(j)}$ and $\mathbf{S}^{(j)}$ are defined as follows:

$$\widehat{\mathbf{S}}^{(j)} = \sum_{l=j}^{\infty}\widehat{\mathbf{F}}^{(l)}\mathbf{G}^{l-j}, \quad j \geq 1, \qquad \mathbf{S}^{(j)} = \sum_{l=j}^{\infty}\mathbf{F}^{(l)}\mathbf{G}^{l-j}, \quad j \geq 0 \ \ (\text{letting } \mathbf{F}^{(0)} \equiv \mathbf{L}). \tag{3}$$

Given the above definition of $\boldsymbol{\pi}^{(j)}$ and the normalization condition, a unique vector $\boldsymbol{\pi}^{(0)}$ can be obtained by solving the following system of linear equations:

$$\boldsymbol{\pi}^{(0)}\left[\left(\widehat{\mathbf{L}}^{(0)} - \widehat{\mathbf{S}}^{(1)}\mathbf{S}^{(0)^{-1}}\widehat{\mathbf{B}}\right)^{\diamond} \ \middle| \ \mathbf{1}^T - \left(\sum_{j=1}^{\infty}\widehat{\mathbf{S}}^{(j)}\right)\left(\sum_{j=0}^{\infty}\mathbf{S}^{(j)}\right)^{-1}\mathbf{1}^T\right] = [\mathbf{0} \mid 1]. \tag{4}$$

where the symbol "$\diamond$" indicates that we discard one (any) column of the corresponding matrix, since we added a column representing the normalization condition. Once $\boldsymbol{\pi}^{(0)}$ is known, we can then iteratively compute $\boldsymbol{\pi}^{(i)}$ for $i \geq 1$, stopping when the accumulated probability mass is close to one. After this point, measures of interest can be computed. [7] gives an improved version of Ramaswami's formula, where the stationary probability vector is computed using matrix-generating functions associated with triangular Toeplitz matrices. These matrix-generating functions are computed efficiently by using fast Fourier transforms (FFT). This algorithm is outlined as follows:

$$\tilde{\boldsymbol{\pi}}^{(1)} = -\mathbf{Y}^{-1}\cdot\mathbf{b}, \qquad \tilde{\boldsymbol{\pi}}^{(i)} = -\mathbf{Y}^{-1}\mathbf{Z}\cdot\tilde{\boldsymbol{\pi}}^{(i-1)} \quad i \geq 2, \tag{5}$$

where the following definitions hold:

$$\tilde{\boldsymbol{\pi}}^{(1)} = [\boldsymbol{\pi}^{(1)}, ..., \boldsymbol{\pi}^{(p)}], \qquad \tilde{\boldsymbol{\pi}}^{(i)} = [\boldsymbol{\pi}^{(pi+1)}, ..., \boldsymbol{\pi}^{(p(i+1))}] \quad i \geq 2$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{S}^{(1)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{S}^{(2)} & \mathbf{S}^{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{S}^{(3)} & \mathbf{S}^{(2)} & \mathbf{S}^{(1)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{S}^{(p)} & \mathbf{S}^{(p-1)} & \mathbf{S}^{(p-2)} & \cdots & \mathbf{S}^{(1)} \end{bmatrix} \quad \mathbf{Z} = \begin{bmatrix} \mathbf{0} & \mathbf{S}^{(p)} & \cdots & \mathbf{S}^{(3)} & \mathbf{S}^{(2)} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S}^{(4)} & \mathbf{S}^{(3)} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S}^{(5)} & \mathbf{S}^{(4)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \widehat{\mathbf{S}}^{(1)} \\ \widehat{\mathbf{S}}^{(2)} \\ \widehat{\mathbf{S}}^{(3)} \\ \vdots \\ \widehat{\mathbf{S}}^{(k)} \end{bmatrix}\boldsymbol{\pi}^{(0)} \tag{6}$$

where p is a constant that defines how many of matrices $\mathbf{S}^{(i)}$ and $\widehat{\mathbf{S}}^{(i)}$ are computed. In the above representation, the matrix $\mathbf{Y}$ is a lower block triangular Toeplitz matrix and the matrix $\mathbf{Z}$ is an upper block triangular Toeplitz matrix.

2.2 GI/M/1-type Processes

The continuous GI/M/1-type Markov processes, with infinitesimal generator of the form $\mathbf{Q}_{GI/M/1}$ admit the very-elegant matrix-geometric solutions proposed by Neuts [8]. In matrix-geometric solutions the relation between the stationary probability vectors $\boldsymbol{\pi}^{(i)}$ is:

$$\boldsymbol{\pi}^{(i)} = \boldsymbol{\pi}^{(1)}\cdot\mathbf{R}^{i-1}, \qquad \forall i \geq 1 \tag{7}$$

where $\mathbf{R}$ is the solution of the matrix equation

$$\mathbf{F} + \mathbf{R} \cdot \mathbf{L} + \sum_{k=1}^{\infty} \mathbf{R}^{k+1} \cdot \mathbf{B}^{(k)} = \mathbf{0}. \tag{8}$$

The matrix $\mathbf{R}$ has an important probabilistic interpretation: $\mathbf{R}$ records the expected number of visits to each state in $\mathcal{S}^{(i)}$, starting from each state in $\mathcal{S}^{(i-1)}$, before reentering $\mathcal{S}^{(i-1)}$ [5]. $\mathbf{R}$ can be obtained using iterative methods. Eq.(8) together with the normalization condition are then used to obtain $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$ by solving the following system of linear equations:

$$[\boldsymbol{\pi}^{(0)},\ \boldsymbol{\pi}^{(1)}] \cdot \left[\begin{array}{c|c|c} \widehat{\mathbf{L}}^{\diamond} & \widehat{\mathbf{F}}^{(1)} & \mathbf{1}^T \\ (\sum_{k=1}^{\infty} \mathbf{R}^{k-1} \cdot \widehat{\mathbf{B}}^{(k)})^{\diamond} & \mathbf{L} + \sum_{k=1}^{\infty} \mathbf{R}^{k} \cdot \mathbf{B}^{(k)} & (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1}^T \end{array} \right] = [\mathbf{0} \mid 1]. \tag{9}$$

For $i > 1$, $\boldsymbol{\pi}^{(i)}$ can be obtained numerically from Eq.(7). More importantly, closed form formulas exist for the calculation of many useful performance metrics, such as the expected number of jobs in the service node (i.e., the system queue length) which is given by $\boldsymbol{\pi}^{(1)}(\mathbf{I} - \mathbf{R})^{-2}$.

2.3 QBD Processes

The intersection of GI/M/1 and M/G/1 type processes result on quasi birth-death process, whose infinitesimal generator is of the form $\mathbf{Q}_{QBD}$. Although QBDs can be solved using the algorithm for either the M/G/1-type or GI/M/1-type process, they are usually solved using the very elegant matrix-geometric approach for GI/M/1-type processes [8]. In this case, Eq.(8) reduces to the matrix quadratic equation

$$\mathbf{F} + \mathbf{R} \cdot \mathbf{L} + \mathbf{R}^2 \cdot \mathbf{B} = \mathbf{0}. \tag{10}$$

The most efficient solution of Eq.(10) is provided by the logarithmic reduction algorithm [5]. $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\pi}^{(1)}$ are obtained by solving the following system of linear equations:

$$[\boldsymbol{\pi}^{(0)},\ \boldsymbol{\pi}^{(1)}] \cdot \left[\begin{array}{c|c|c} \widehat{\mathbf{L}}^{\diamond} & \widehat{\mathbf{F}} & \mathbf{1}^T \\ \widehat{\mathbf{B}}^{\diamond} & \mathbf{L} + \mathbf{R} \cdot \mathbf{B} & (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1}^T \end{array} \right] = [\mathbf{0} \mid 1]. \tag{11}$$

Similarly to GI/M/1 processes, several measures of interest have closed form formulas that are based on $\boldsymbol{\pi}^{(0)}$, $\boldsymbol{\pi}^{(1)}$ and $\mathbf{R}$ only.

2.4 ETAQA Methodology

[12, 2, 3] propose a methodology to compute only aggregate probabilities instead of the whole stationary probability vector for M/G/1 and QBD processes. This aggregate methodology, known as ETAQA is exact, numerically stable and most importantly very efficient with respect to both its time and space complexity. For an ergodic CTMC, it requires only the knowledge of matrix $\mathbf{G}$. The aggregate solution of an M/G/1 process is obtained as the solution of the following system of linear equations:

$$[\boldsymbol{\pi}^{(0)},\ \boldsymbol{\pi}^{(1)},\ \boldsymbol{\pi}^{(*)}] \cdot \left[\begin{array}{c|c|c|c} \mathbf{1}^T & \widehat{\mathbf{L}} & \widehat{\mathbf{F}}^{(1)} - \sum_{i=3}^{\infty} \widehat{\mathbf{S}}^{(i)} \cdot \mathbf{G} & (\sum_{i=2}^{\infty} \widehat{\mathbf{F}}^{(i)} + \sum_{i=3}^{\infty} \widehat{\mathbf{S}}^{(i)} \cdot \mathbf{G})^{\diamond} \\ \mathbf{1}^T & \widehat{\mathbf{B}} & \mathbf{L} - \sum_{i=2}^{\infty} \mathbf{S}^{(i)} \cdot \mathbf{G} & (\sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \sum_{i=2}^{\infty} \mathbf{S}^{(i)} \cdot \mathbf{G})^{\diamond} \\ \mathbf{1}^T & \mathbf{0} & \mathbf{B} - \sum_{i=1}^{\infty} \mathbf{S}^{(i)} \cdot \mathbf{G} & (\sum_{i=1}^{\infty} \mathbf{F}^{(i)} + \mathbf{L} + \sum_{i=1}^{\infty} \mathbf{S}^{(i)} \cdot \mathbf{G})^{\diamond} \end{array} \right] = [1, \mathbf{0}], \tag{12}$$

where $\boldsymbol{\pi}^{(*)}$ is the aggregate probability and is defined as $\boldsymbol{\pi}^{(*)} = \sum_{i=2}^{\infty} \boldsymbol{\pi}^{(i)}$.

As a special case of M/G/1 processes, QBDs can be solved using this aggregate approach by solving the following simplified system of linear equations:

$$[\boldsymbol{\pi}^{(0)},\ \boldsymbol{\pi}^{(1)},\ \boldsymbol{\pi}^{(*)}] \cdot \left[\begin{array}{c|c|c|c} \mathbf{1}^T & \widehat{\mathbf{L}} & \widehat{\mathbf{F}} & \mathbf{0}^{\diamond} \\ \mathbf{1}^T & \widehat{\mathbf{B}} & \mathbf{L} & \mathbf{F}^{\diamond} \\ \mathbf{1}^T & \mathbf{0} & \mathbf{B} - \mathbf{F} \cdot \mathbf{G} & (\mathbf{L} + \mathbf{F} + \mathbf{F} \cdot \mathbf{G})^{\diamond} \end{array} \right] = [1, \mathbf{0}], \tag{13}$$

[12, 2, 3] provide algorithms for computation of measures of interests based on the computed aggregate probability $[\boldsymbol{\pi}^{(0)}, \ \boldsymbol{\pi}^{(1)}, \ \boldsymbol{\pi}^{(*)}]$.

3 MAMSolver

MAMSolver is a tool that provides implementation of the algorithms highlighted in Section 2. The tool provides solutions for both DTMCs and CTMCs. The matrix-analytic algorithms that we take under consideration are defined in terms of matrices, making matrix manipulations and operations the basic elements of the tool. The input to MAMSolver, in the form of a structured text file, is the finite set of matrices that accurately describe the process to be solved. Since there are different algorithms that provide solution for the same process, the user specifies the method to be used. However, several tests are performed within the tool to insure that special cases are treated separately and therefore more efficiently. MAMSolver is implemented in C++, and classes define the basic components of the type of processes under consideration.

Matrix is the module that implements all the basic matrix operations such as matrix assignments, additions, subtractions, multiplications, and inversions. For computational efficiency, we use well known and heavily tested routines provided by the Lapack and BLAS packages[2]. Since solving a finite system of linear equations is a core function in matrix-analytic algorithms, MAMSolver provides several numerical methods depending on the size of the problem, i.e., the size of the coefficient matrices. For small-size problems exact methods such as LU-decomposition are used, otherwise the Lapack implementation of iterative methods such as GMRES and BiCGSTAB, are chosen.

Matrix-analytic modules handle both CTMC and DTMC processes. First these modules provide storage for the input matrices. In addition, these modules provide all the routines necessary for the implementation of the algorithms outlined in Section 2. Both the data structures and routines of the matrix-analytic modules are based on the data-structures and routines provided by the matrix module.

The solution of **QBD** processes, requires computation of $\mathbf{R}$ (and sometimes of $\mathbf{G}$ depending on the solution algorithm). First matrix $\mathbf{R}$ is computed using the logarithmic reduction algorithm[5]. For completeness we provide also the classical numerical algorithm. To guarantee that there is no expensive (and unnecessary) iterative computation of $\mathbf{G}$ (and $\mathbf{R}$), the tool first checks if the conditions for explicit computation hold [11]. The available solution methods for QBD processes are matrix-geometric and ETAQA.

GI/M/1 processes require the computation of matrix $\mathbf{R}$. The classic matrix geometric solution is implemented to solve this type of processes. First the algorithm goes through a classic iterative algorithm to compute $\mathbf{R}$ (to our knowledge, there is no alternative more efficient than the classic algorithm). Then, the tool computes the boundary part of the stationary probability vector. Since a geometric relation exist between vectors $\boldsymbol{\pi}^{(i)}$ for $i \geq 1$, there is no need to compute the whole stationary probability vector.

M/G/1 processes require the computation of matrix $\mathbf{G}$ which is calculated using the classic iterative algorithm or the cyclic-reduction algorithm or the explicit one (if applied). The stationary probability vector is computed recursively using either the recursive Ramaswami formula or its fast FFT version. ETAQA is also implemented as an alternative for the solution of M/G/1 processes.

MAMsolver computes and stores the stationary probability vector, allowing the user to compute any measures of interest. Momentarily MAMSolver computes a customized first moment of the form:

$$r = \boldsymbol{\pi}^{(0)} \cdot \boldsymbol{\rho}^{(0)} + \boldsymbol{\pi}^{(1)} \cdot \boldsymbol{\rho}^{(1)} + \sum_{i=2}^{infty} \boldsymbol{\pi}^{(i)}(\mathbf{a}^{[0]} + \mathbf{a}^{[1]} \cdot i)$$

All the coefficient vectors $\boldsymbol{\rho}^{(0)}$, $\boldsymbol{\rho}^{(1)}$, $\mathbf{a}^{[0]}$, $\mathbf{a}^{[1]}$ are to be read from a text file with default values: $\boldsymbol{\rho}^{(0)} = \mathbf{0}$, $\boldsymbol{\rho}^{(1)} = \mathbf{1}$, $\mathbf{a}^{[0]} = \mathbf{0}$, $\mathbf{a}^{[1]} = \mathbf{1}$. Computation of higher moments can be achieved by introducing additional coefficient vectors $\mathbf{a}^{[j]}$ in the above formula.

More details and the whole source code for MAMSolver is available at http://www.cs.wm.edu/˜riska/MAMSolver.html.

[2] Available from http://www.netlib.org.

4 Future directions

We aim to extend the set of measures of interest that are computed by MAMSolver and increase the level of abstraction such that the user can define them based on their needs. Further, we plan to link this tool with another set of modules that we have on fitting data into distributions such as BMAP, MAP, PH. These type of distributions accurately capture the characteristics of network and Internet traffic and the corresponding queuing system fall into the cases of M/G/1, GI/M/1 and QBD processes. We plan to provide a framework that uses Markovian modeling and additional analytic methods to analyse network and telecommunication traffic.

Acknowledgments

We thank Beatrice Meini for providing us with her Fortran implementation of the cyclic reduction algorithm for computation of matrix **G** and the fast FFT Ramaswami's formula. We also thank Daniela Puiu for her contribution in earlier versions of the tool.

References

[1] D.A. Bini and B. Meini. Using displacement structure for solving non-skip-free M/G/1 type Markov chains. Advances in Matrix Analytic Methods for Stochastic Models - Proceedings of the 2nd international conference on matrix analytic methods, A. Alfa and S. Chakravarthy Eds., 1998, Notable Publications Inc, NJ, pages 17-37.

[2] G. Ciardo, A. Riska, and E. Smirni. An aggregation-based solution method for M/G/1-type processes. In B. Plateau, W. J. Stewart, and M. Silva, editors, *Numerical Solution of Markov Chains*, pages 21–40. Prensas Universitarias de Zaragoza, Zaragoza, Spain, Sept. 1999.

[3] G. Ciardo and E. Smirni. ETAQA: an efficient technique for the analysis of QBD-processes by aggregation. *Performance Evaluation*, Vol.(36-37), pages 71–93, 1999.

[4] B.R. Haverkort, A.P.A. van Moorsel and A. Dijkstra. MGMtool: A Performance Analysis Tool Based on Matrix Geometric Methods. In R. Pooley, and J. Hillston, editors *Modelling Techniques and Tools*, pages 312–316, Edinburgh University Press, 1993.

[5] G. Latouche and V. Ramaswami. *Introduction to Matrix Geometric Methods in Stochastic Modeling.* ASA-SIAM Series on Statistics and Applied Probability. SIAM, Philadelphia PA, 1999.

[6] D. M. Lucantoni. The BMAP/G/1 queue: A tutorial. In L. Donatiello and R. Nelson, editors, *Models and Techniques for Performance Evaluation of Computer and Communication Systems*, pages 330–358. Springer-Verlag, 1993.

[7] B. Meini. An improved FFT-based version of Ramaswami's formula. *Comm. Statist. Stochastic Models*, vol. 13, pages 223-238, 1997.

[8] M. F. Neuts. *Matrix-geometric solutions in stochastic models.* Johns Hopkins University Press, Baltimore, MD, 1981.

[9] M. F. Neuts. *Structured stochastic matrices of M/G/1 type and their applications.* Marcel Dekker, New York, NY, 1989.

[10] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Commun. Statist.- Stochastic Models*, vol. 4 pages 183-263, 1988.

[11] V. Ramaswami and G. Latouche. A general class of Markov processes with explicit matrix-geometric solutions. *OR Spektrum*, 8:209–218, Aug. 1986.

[12] A. Riska, and E. Smirni. An exact aggregation approach for M/G/1-type Markov chains. submitted for publication.

[13] Mark.S.Squillante. MAGIC: A computer performance modeling tool based on matrix-geometric techniques. In Proc. *Fifth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, 1991.

[14] M. S. Squillante. Matrix-analytic methods: Applications, results and software tools. In *Advances in Matrix-Analytic Methods for Stochastic Models,* G. Latouche and P. Taylor (eds.). Notable Publications, 2000.