Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 185 (2008) 216–234
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Stochastics and Statistics

Bounds and approximations for continuous-time Markovian transition probabilities and large systems

Sophie Mercier *

Laboratoire d'Analyse et de Mathématiques Appliquées (CNRS - UMR 8050), Université de Marne-la-Vallée, Cité Descartes, 5 boulevard Descartes, Champs sur Marne, 77454 Marne-la-Vallée Cedex 2, France

Received 15 March 2006; accepted 31 December 2006
Available online 30 January 2007

Abstract

We propose new bounds and approximations for the transition probabilities of a continuous-time Markov process with finite but large state-space. The bounding and approximating procedures have been exposed in another paper [S. Mercier, Numerical bounds for semi-Markovian quantities and applications to reliability, in revision for Methodology and Computing in Applied Probability] in the more general context of a continuous-time semi-Markov process with countable state-space. Such procedures are here specialized to the Markovian finite case, leading to much simpler algorithms. The aim of this paper is to test such algorithms versus other algorithms from the literature near from ours, such as forward Euler approximation, external uniformization and a finite volume method from [C. Cocozza-Thivent, R. Eymard, Approximation of the marginal distributions of a semi-Markov process using a finite volume scheme, ESAIM: M2AN 38(5) (2004) 853–875].
© 2007 Elsevier B.V. All rights reserved.

Keywords: Markov processes; Numerical bounds and approximations; Time-dependent quantities; Reliability

1. Introduction

Continuous-time jump Markov processes with finite state-space are very common for modelling the time evolution of industrial systems. However, due to a rapid explosion of the size of the state-space with the number of components of the system, the numerical evaluation of different reliability indicators might become a challenging problem. Such considerations have lead to an extensive literature on the evaluation of such indicators and especially of the transition probabilities from which may be derived lots of other time-dependent quantities: to have an idea of the existing literature, one may look for instance at the book by Stewart [16], to the articles by Moler and Van Loan [10,11] with 161 citations, to [13] or to [15]. The aim of this paper is not to do any survey of the subject neither to look at all the existing methods such as uniformization, Krylov subspace techniques, ODE methods or any other ones or any of their derivatives. We here simply present a new

* Tel.: +33 1 60 95 75 40; fax: +33 1 60 95 75 45.
E-mail address: Sophie.Mercier@univ-mlv.fr

0377-2217/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2006.12.036

method which allows to bound and to approximate the goal quantities, namely the transition probabilities of a continuous-time Markov process with a large finite state-space. This method is also compared to three other ones from the literature, chosen for their proximity with ours.

The principle is very simple and consists in bounding the continuous inter-arrival times of a jump Markov process by two discrete random variables with range in $h\mathbb{N}$ ($h > 0$). This bounding procedure is valid for any general random variable and has already been used in [8] to bound different quantities from the reliability field such as (1) cumulative probability functions of sums of i.i.d. non-negative random variables, (2) renewal functions and (3) cumulative probability functions of geometric sums of i.i.d. non-negative random variables. More generally, it has been used in [9] to bound any quantity which is solution of a Markov renewal equation linked to some semi-Markov process with countable state-space, and in particular to bound the transition probabilities of a semi-Markov process. The present paper may then be seen as a special case of [9]. However, in the present case of a Markov process, the algorithm for the computations of the bounds may be highly simplified leading to much quicker computations. Also, the approximations provided by the bounds for the transition probabilities may here be interpreted as first-order expansions for the semi-group of the Markov process, just as for well-known methods such as Euler's forward approximation (FEA) [12] or external uniformization (EUA) [6] or [14], or for the method provided in [4] for semi-Markov processes (CEA). Those three methods (FEA, EUA and CEA) have then been chosen for comparison purpose with ours due to their proximity from ours.

This paper is organized as follows: the notations and the mathematical backgrounds from [9] are given in Section 2. Algorithms are provided in Section 3 for the numerical computations of bounds and approximations of any solution of a Markov renewal equation associated to some continuous-time Markov process. Section 4 is specialized to the bounds and approximations of the transition probabilities and to their interpretation in term of the continuous-time Markov process. A benchmark is provided in Section 5 which is used in Section 6 to perform different numerical experiments: the bounds are first tested. Then, different approximations from the paper are compared and a single one is selected to be next compared to FEA, EUA and CEA. Conclusive remarks close the paper in Section 7.

2. Notations and mathematical backgrounds

A system is considered, which evolves in time according to a continuous time Markov process $(X_t)_{t\geqslant 0}$ on a finite state-space $E = \{1, \ldots, m\}$. For $i, j \in E$ with $i \neq j$, let $a_{i,j}$ be the constant transition rate from state i to state j and let $b_i = \sum_{j\neq i} a_{i,j}$. Also, let A be the generator matrix of $(X_t)_{t\geqslant 0}$ with:

$$A(i,j) = \begin{cases} a_{i,j} & \text{if } i \neq j, \\ -b_i & \text{if } i = j \end{cases}$$

and let P be the transition matrix with:

$$P_{i,j} = \begin{cases} \frac{a_{i,j}}{b_i} & \text{if } b_i \neq 0, \\ 0 & \text{if } b_i = 0 \end{cases}$$

for $i \neq j$ and $P_{i,i} = 0$.

We also set $T_0 = 0 < T_1 < \cdots < T_n < \cdots$ as the jump-times for $(X_t)_{t\geqslant 0}$ with $\sup_{n\in\mathbb{N}} T_n = +\infty$ a.s. for all $i \in E$ and $(P_t)_{t\geqslant 0}$ the transition semi-group of $(X_t)_{t\geqslant 0}$:

$$P_t(i,j) = \mathbb{P}_i(X_t = j)$$

for all $i, j \in E$, all $t \geqslant 0$ where $\mathbb{P}_i$ is the conditional distribution given that $X_0 = i$.

Now, let $\mathbb{B}_+$ be the set of all functions $f : E \times \mathbb{R}_+ \to \mathbb{R}_+$ such that the function $t \mapsto f(i,t)$ is bounded on all $[0,t]$ for all $i \in E$.

Letting $q(i,j,\mathrm{d}u) = a_{i,j}\mathrm{e}^{-b_i u}\mathrm{d}u$ (all $i, j \in E$), we recall that for any fixed $g \in \mathbb{B}_+$, the Markov renewal equation $f = g + q * f$, namely,

$$f(i,t) = g(i,t) + \sum_{j\in E} \int_0^t a_{i,j} \mathrm{e}^{-b_i u} f(j, t-u)\,\mathrm{d}u \quad (\text{all } i \in E,\ \text{all } t \geqslant 0)$$

has got one single solution in $\mathbb{B}_+$ (see [2] or [3], e.g.), which we denote by f_g.

In [9], it has been proved that such a single solution f_g may be approximated by some functions f_g^h and f_g^{h+}. Such functions are the single solutions of discrete Markov renewal equations: $f_g^h = g + q^h * f_g^h$ and $f_g^{h+} = g + q^{h+} * f_g^{h+}$ where the associated discrete semi-Markov kernels $(q^h(i,j,\mathrm{d}t))_{i,j\in E}$ and $(q^{h+}(i,j,\mathrm{d}t))_{i,j\in E}$ have a support included in $h\mathbb{N}$ and where $q^h * f$ stands for

$$(q^h * f)(i,Nh) = \sum_{j\in E} \sum_{k=0}^{N} q^h(i,j,kh) f(j,(N-k)h), \tag{1}$$

for all $f \in \mathbb{B}_+$, the same for $q^{h+} * f$.

The approximating discrete time semi-Markov processes associated to $(q^h(i,j,\mathrm{d}t))_{i,j\in E}$ and $(q^{h+}(i,j,\mathrm{d}t))_{i,j\in E}$ are constructed such that they visit the same states as the initial continuous-time Markov process (CTMP) and such that their respective inter-arrival times are the classical lower and upper geometric approximations for the exponential inter-arrival times of the CTMP. The discrete semi-Markov kernels then have the following mass at kh:

$$q^h(i,j,kh) = \mathbb{P}_i(kh < T_1 \leqslant (k+1)h, X_{T_1} = j) = P_{i,j}\mathrm{e}^{-b_i kh}(1-\mathrm{e}^{-b_i h}) \tag{2}$$

$$q^{h+}(i,j,kh) = \mathbf{1}_{\{k\geqslant 1\}} q^h(i,j,(k-1)h) = \mathbf{1}_{\{k\geqslant 1\}} P_{i,j}\mathrm{e}^{-(k-1)b_i h}(1-\mathrm{e}^{-b_i h}) \tag{3}$$

for all $k \in \mathbb{N}$ where $\mathbf{1}_{\{.\}}$ is the indicator function.

The upper semi-Markov approximation may also be considered as a Markov chain with time scale $h\mathbb{N}^*$ and transition probabilities:

$$\pi_{i,j} = \begin{cases} q^{h+}(i,j,h) = P_{i,j}(1-\mathrm{e}^{-b_i h}) & \text{for } i \neq j, \\ 1 - \sum_{j\neq i} q^{h+}(i,j,h) = \mathrm{e}^{-b_i h} & \text{for } i = j. \end{cases}$$

This is not the case for the lower semi-Markov approximation which may instantaneously jump with non-zero probability (due to $q^h(i,j,0) = P_{i,j}(1-\mathrm{e}^{-b_i h})$).

Due to the construction, the approximations $f_{g_j}^h$ and $f_{g_j}^{h+}$ will be referred to as LGA and UGA in the following, for "lower" and "upper geometric approximation", respectively.

The following results have been proved in [9] (in more general a context).

Theorem 1. *Let $(X_t)_{t\geqslant 0}$ be a Markov process with a finite state-space. For all $g \in \mathbb{B}_+$:*

(1) *if $t \mapsto g(i,t)$ is non-decreasing for all $i \in E$, then, for all $0 < h$:*

$$f_g^{h+} \leqslant f_g \leqslant f_g^h < +\infty, \tag{4}$$

(2) *if g is of the shape $g = g_1 - g_2$ with $g_1, g_2 \in \mathbb{B}_+$ and $t \mapsto g_j(i,t)$ is non-decreasing for $j = 1,2$ and all $i \in E$, then, for all $0 < h$:*

$$f_{g_1}^{h+} - f_{g_2}^h \leqslant f_g = f_{g_1} - f_{g_2} \leqslant f_{g_1}^h - f_{g_2}^{h+} < +\infty, \tag{5}$$

(3) *if $u \mapsto g(i,u)$ is uniformly continuous on $[0,t]$ for all $i \in E$ where $t \geqslant 0$, then*

$$\lim_{h\to 0^+} f_g^h(i,t) = \lim_{h\to 0^+} f_g^{h+}(i,t) = f_g(i,t). \tag{6}$$

In some special cases, we can also use the following proposition proved in [9] (in more general a context):

Proposition 2. *Let $(X_t)_{t\geqslant 0}$ be a Markov process on $E = \{1,\ldots,m\}$ and $v : E \to \mathbb{R}_+$. Let $w : E \times \mathbb{R}_+ \to \mathbb{R}_+$ be such that $w(i,t) = v(i)\mathbb{P}_i(T_1 > t) = v(i)e^{-b_i t}$. If the generator matrix A of $(X_t)_{t\geqslant 0}$ is upper triangular and if v is non-decreasing, we have, for all $h > 0$, $t \geqslant 0$, $i \in E$:*

$$f^{h+}_{w^{h+}}(i,t) \leqslant \mathbb{E}_i(v(X_t)) \leqslant f^{h}_{w^{h}}(i,t) \tag{7}$$

with

$$w^h(i,t) = v(i)\exp\left(-b_i\left(h\left\lfloor\frac{t}{h}\right\rfloor + h\right)\right)$$
$$w^{h+}(i,t) = v(i)\exp\left(-b_i h\left\lfloor\frac{t}{h}\right\rfloor\right),$$

where $\lfloor\cdots\rfloor$ *stands for the floor function. (For* $x \in \mathbb{R}$ *and* $n \in \mathbb{Z} : \lfloor x\rfloor = n$ *if and only if* $n \leqslant x < n+1$*). If A is lower triangular, inequalities are reversed; the same if v is non-increasing.*

Inequalities (4), (5) and (7) provide bounds for the goal quantity f_g which converge to f_g due to (6). The interest of such results is that the bounds are easily computable. The following section is devoted to their numerical computation in the present case of a Markov process.

3. Numerical computation of f_g and f_g^{h+}

Using (1), the equation $f_g^h = g + q^h * f_g^h$ (e.g.) may be written at $t = Nh$ as

$$f_g^h(i,Nh) = g(i,Nh) + \sum_{j\in E}\sum_{k=0}^{N} q^h(i,j,kh) f_g^h(j,(N-k)h) \tag{8}$$

(similar result for f_g^{h+}) so that $f_g^h(\cdot,Nh)$ may easily be expressed with respect of $f_g^h(\cdot,0\times h)$, $f_g^h(\cdot,1\times h),\ldots,f_g^h(\cdot,(N-1)h)$, where $f^h(\cdot,kh)$ stands for $(f^h(i,kh))_{i\in E}$ (all $0 \leqslant k \leqslant N$), see [9]. However, in the present case where $(X_t)_{t\geqslant 0}$ is a Markov process, we may express $f_g^h(\cdot,Nh)$ only with respect of $f_g^h(\cdot,(N-1)h)$ and of the initial data (the same for f_g^{h+}), leading to much quicker a computation. We get the following result:

Proposition 3. *Let* $g \in \mathbb{B}_+$ *and* $h > 0$*. For all* $N \in \mathbb{N}$*, we set:* $f^h(\cdot,Nh) = (f^h(i,Nh))_{i\in E}$ *columnwise, the same for* $g(\cdot,Nh)$ *and* $f^{h+}(\cdot,Nh)$*. Let I be the identity matrix with size cardinal*(E) *and*

$$D_h = \mathrm{diag}(\mathrm{e}^{-b_i h},\ i = 1\ldots m)$$
$$C_h = D_h + (I - D_h)P$$
$$B_h = I - (C_h - D_h) = I - (I - D_h)P,$$

where $\mathrm{diag}(u_1,\ldots,u_m)$ *stands for the diagonal matrix with* $u_1,\ldots,u_m$ *as diagonal terms (all* $u_1,\ldots,u_m \in \mathbb{R}$*). The matrix* B_h *then is non-singular and*

$$f_g^h(\cdot,0) = B_h^{-1} g(\cdot,0), \tag{9}$$
$$f_g^{h+}(\cdot,0) = g(\cdot,0), \tag{10}$$
$$f_g^h(\cdot,(N+1)h) = B_h^{-1}[g(\cdot,(N+1)h) + D_h(f_g^h(\cdot,Nh) - g(\cdot,Nh))], \tag{11}$$
$$f_g^{h+}(\cdot,(N+1)h) = C_h f_g^{h+}(\cdot,Nh) - D_h g(\cdot,Nh) + g(\cdot,(N+1)h), \tag{12}$$

for all $N \in \mathbb{N}$.

Proof. Using (2) and (3), we may write

$$q^h(\cdot,\cdot,kh) = D_h^k(I - D_h)P = D_h^k(C_h - D_h),$$
$$q^{h+}(\cdot,\cdot,kh) = \mathbf{1}_{\{k\geqslant 1\}} D_h^{k-1}(C_h - D_h),$$

where $q^h(\cdot,\cdot,kh) = (q^h(i,j,kh))_{i,j\in E}$, the same for $q^{h+}(\cdot,\cdot,kh)$.

Eq. (8) may now be written as

$$f_g^h(\cdot,Nh) = g(\cdot,Nh) + \sum_{k=0}^{N} D_h^{N-k}(C_h - D_h) f_g^h(\cdot,kh),$$

for all $N \in \mathbb{N}$. Whence,

$$(I-(C_h-D_h))f_g^h(\cdot,Nh) = B_h f_g^h(\cdot,Nh) = g(\cdot,Nh) + \mathbf{1}_{\{N\geqslant 1\}}\sum_{k=0}^{N-1} D_h^{N-k}(C_h-D_h)f_g^h(\cdot,kh).$$

Now, let λ be an eigenvalue of $C_h - D_h = (I-D_h)P$. Using $\sum_{j\in E}P_{k,j}=1$ for all $k\in E$, we have

$$|\lambda| \leqslant \|(I-D_h)P\|_\infty = \max_{i\in E}\left(\sum_{j\in E}[(I-D_h)P](i,j)\right) = \max_{i\in E}((1-D_h)(i,i)) = \max_{i\in E}(1-\mathrm{e}^{-hb_i}) < 1 \tag{13}$$

and 1 is not an eigenvalue of $(I-D_h)P$. Then, $B_h = I-(C_h-D_h)$ is non-singular. We derive

$$f_g^h(\cdot,Nh) = B_h^{-1}\left[g(\cdot,Nh) + \mathbf{1}_{\{N\geqslant 1\}}\sum_{k=0}^{N-1} D_h^{N-k}(I-B_h)f_g^h(\cdot,kh)\right] \tag{14}$$

and consequently, $f_g^h(\cdot,0) = B_h^{-1}g(\cdot,0)$.

For any $N\in\mathbb{N}$, we also have from (14) at $N+1$:

$$\begin{aligned}
f_g^h(\cdot,(N+1)h) &= B_h^{-1}[g(\cdot,(N+1)h) + \sum_{k=0}^{N} D_h^{N+1-k}(I-B_h)f_g^h(\cdot,kh)] \\
&= B_h^{-1}\begin{bmatrix} g(\cdot,(N+1)h) + D_h(I-B_h)f_g^h(\cdot,Nh) \\ +\mathbf{1}_{\{N\geqslant 1\}}\sum_{k=0}^{N-1} D_h^{N+1-k}(I-B_h)f_g^h(\cdot,kh)\end{bmatrix} \\
&= B_h^{-1}\begin{bmatrix} g(\cdot,(N+1)h) \\ +D_h(f_g^h(\cdot,Nh) - (B_h f_g^h(\cdot,Nh) - \mathbf{1}_{\{N\geqslant 1\}}\sum_{k=0}^{N-1} D_h^{N-k}(I-B_h)f_g^h(\cdot,kh)))\end{bmatrix} \\
&= B_h^{-1}[g(\cdot,(N+1)h) + D_h(f_g^h(\cdot,Nh) - g(\cdot,Nh))]
\end{aligned}$$

from (14) at N, whence the result for f_g^h. The proof is similar for f_g^{h+} and is omitted. □

We easily derive the following result:

Corollary 4. *With the assumptions and notations of Proposition 3, in case $g(\cdot,(N+1)h) = D_h \times g(\cdot,Nh)$ for all $N\in\mathbb{N}$, we then have*

$$f_g^h(\cdot,(N+1)h) = B_h^{-1}D_h f_g^h(\cdot,Nh), \tag{15}$$

$$f_g^{h+}(\cdot,(N+1)h) = C_h f_g^{h+}(\cdot,Nh) \tag{16}$$

and consequently

$$f_g^{h+}(\cdot,Nh) = C_h^N \times g(\cdot,0), \tag{17}$$

$$f_g^h(\cdot,Nh) = (B_h^{-1}D_h)^N B_h^{-1} \times g(\cdot,0), \tag{18}$$

for all $N\in\mathbb{N}$.

Remark 5. The previous corollary is used in Section 4.2 for the interpretation of the approximation of the transition probabilities in term of the underlying continuous-time Markov process.

We can see from Proposition 3 and Corollary 4 that quantities of the shape $B_h^{-1}v$ have to be computed for the numerical evaluation of $f_g^h(\cdot,Nh)$, with v a column array. This is equivalent to the resolution of linear systems of the shape $B_h w = v$. The conditioning of such systems is studied in the following lemma:

Lemma 6. *For $h>0$, the condition number of B_h is given by*

$$\mathrm{cond}(B_h) = \mathrm{e}^{h(\max_{i\in E}b_i - \min_{i\in E}b_i)},$$

where $\mathrm{cond}(B_h) = \|B_h\|_\infty . \|B_h^{-1}\|_\infty$ *with* B_h *defined in Proposition 3 and*

$$\|B_h\|_\infty = \max_{i\in E}\left(\sum_{j\in E} B_h(i,j)\right).$$

Proof. Using similar arguments as in (13), we have

$$\|B_h\|_\infty = \max_{i\in E}\left(\sum_{j\in E}[I-(I-D_h)P](i,j)\right) = \max_{i\in E}(D_h(i,i)) = \mathrm{e}^{-h\min_{i\in E} b_i}.$$

Beside

$$\|B_h^{-1}\|_\infty = \|(I-(I-D_h)P)^{-1}\|_\infty = \|\sum_{k=0}^{+\infty}((I-D_h)P)^k\|_\infty \leqslant \sum_{k=0}^{+\infty}\|((I-D_h)P)\|_\infty^k, = \sum_{k=0}^{+\infty}\left(\max_{i\in E}(1-\mathrm{e}^{-hb_i})\right)^k$$
$$= \frac{1}{\mathrm{e}^{-h\max\limits_{i\in E} b_i}} = \mathrm{e}^{h\max_{i\in E} b_i},$$

hence the result. □

In case of h small in front of $(\max_{i\in E} b_i - \min_{i\in E} b_i)$, the previous lemma shows that the condition number of B_h is near from 1 and systems like $B_h w = v$ are consequently well conditioned (see [1], e.g.).

We now provide an algorithm for the computation of f_g^h and f_g^{h+}. Due to the specialization of the present paper to the computation of the transition probabilities, such an algorithm is here written in the special case where $g(\cdot,(k+1)h) - D_h g(\cdot,kh)$ is independent of k, which is true for the quantities of interest (see Section 4). Note that this property is actually true for lots of other quantities such as mean cumulated sojourn times on $[0,t]$, mean number of jumps from one state to another state on $[0,t]$, ... see [7] for details.

To write our algorithm, we use the initialization (9–10), the recursive formulas (11–12), and the assumption $g(\cdot,(k+1)h) - D_h g(\cdot,kh)$ independent on k. Also, to accelerate the computation and save memory, we use the following two "tricks":

- All values for $f_g^h(\cdot,kh)$ and $f_g^{h+}(\cdot,kh)$ with $0 \leqslant k \leqslant N$ are generally not required. It is then unnecessary to remember all $f_g^h(\cdot,kh)$ and $f_g^{h+}(\cdot,kh)$, which is memory and time consuming. We here consider the case where only one out of n_0 values are required and we only retain the values $f_g(\cdot,n_0h)$, $f_g(\cdot,2n_0h),\ldots,f_g(\cdot,N_0n_0h) = f_g(\cdot,Nh)$ with $N = N_0n_0$ (eventually $N_0 = 1$ and $n_0 = N$ in case only $f_g^h(\cdot,Nh)$ and $f_g^{h+}(\cdot,Nh)$ are required).
- Also, to accelerate the numerical solving of the successive systems of the shape $B_h w = v$ (one at each step), we factorize B_h once for all at the beginning of the algorithm in the shape $B_h = LU$ with L lower triangular and U upper triangular (eventually up to a permutation, see [1], e.g. or the documentation of Matlab). Computing $B_h^{-1}v$ at each step then resumes to the successive solving of $v = Lz$ and $z = Uw$, which is written $B_h^{-1}v = U \setminus (L \setminus v)$ below, using the notations from Matlab.

We get the following algorithm:

Algorithm 1

- Input data: h, N_0, n_0, A, P, $g(\cdot,0)$, $g(\cdot,h)$

  $b \leftarrow -\mathrm{Diag}(A)$(main diagonal of $-A$ written columnwise),

  $D_h \leftarrow \mathrm{diag}(\mathrm{e}^{-hb_i},\ i = 1,\ldots,m)$,

  $E \leftarrow (I-D_h)P$,

  $C_h \leftarrow D_h + E$,

  $B_h \leftarrow I - E$,

  $[L;U] \leftarrow LU$ factorization of B_h (once for all).

- Initialization:

$$f(\cdot,0) \leftarrow U \setminus (L \setminus g(\cdot,0)),$$
$$f^{+}(\cdot,0) \leftarrow g(\cdot,0),$$
$$X \leftarrow g(\cdot,h) - D_h g(\cdot,0).$$

- For $k = 0$ to $N_0 - 1$ do

$$\begin{cases} Y \leftarrow f(\cdot,k) \\ Z \leftarrow f^{+}(\cdot,k) \\ \text{For } l = 1 \text{to } n_0 \text{ do} \\ \quad \begin{cases} Y \leftarrow U \setminus (L \setminus [D_h Y + X]) \\ Z \leftarrow C_h Z + X \end{cases} \\ f(\cdot,k+1) \leftarrow Y \\ f^{+}(\cdot,k+1) \leftarrow Z \end{cases}$$

- Output : $f(i,kh)$ and $f^{+}(i,kh)$, namely, $f_g^h(i,kn_0h)$ and $f_g^{h+}(i,kn_0h)$ for all $i \in E$, all $0 \leqslant k \leqslant N_0$ $(N = N_0 n_0)$.

Note that this algorithm is free from matrix products and requires only products of the type matrix by a column vector. This is very important in case of E with large size which often arises in reliability (see [12], e.g.).

Also, in case $g(\cdot,(N+1)h) = D_h \times g(\cdot,Nh)$ for all $N \in \mathbb{N}$ (examples are provided in Section 4.2), the algorithm may be simplified using (15 and 16) instead of (11 and 12), namely taking $X = 0$ in its formulation.

Finally, in case $g(\cdot,(k+1)h) - D_h g(\cdot,kh)$ is not independent on k, a more general algorithm might easily be written starting from (11 and 12). Such an algorithm might be used to bound most time-dependent quantities for large Markov systems, which generally are solutions of Markov renewal equations (see [7] or [9] for details). Such algorithms in the Markovian case are much simpler and much more performant than the more general one which is presented in [9] for semi-Markovian systems.

4. Bounds and approximations for $P_t(i,j)$

We here indicate how to apply the previous results to get bounds and approximations for the transition probabilities $P_t(i,j) = \mathbb{P}_i(X_t = j)$. We also briefly describe other known approximations near from ours for comparison purpose.

4.1. Bounds

For $j \in E$, let $f_j : E \times \mathbb{R}_+ \to \mathbb{R}_+$ be such that $f_j(i,t) = P_t(i,j)$ for all $i \in E$, all $t \geqslant 0$. Using classical renewal arguments (see [2] or [3], e.g.), it is known that

$$f_j(i,t) = \mathbf{1}_{\{i=j\}} \mathbb{P}_i(T_1 > t) + \sum_{k \in E} \mathbb{P}_i(X_{T_1} = k, X_t = j, T_1 \leqslant t) = \mathbf{1}_{\{i=j\}} \mathrm{e}^{-b_i t} + \sum_{k \in E} \int_0^t a_{i,k} \mathrm{e}^{-b_i u} \mathbb{P}_k(X_{t-u} = j) \mathrm{d}u$$
$$= g_j(i,t) + \sum_{k \in E} \int_0^t f_j(k,t-u) q(i,k,\mathrm{d}u),$$

where $g_j(i,t) = \mathbf{1}_{\{i=j\}} \mathbb{P}_i(T_1 > t) = \mathbf{1}_{\{i=j\}} \mathrm{e}^{-b_i t}$ for all $i \in E$, all $t \geqslant 0$.

Then f_j is solution of the Markov renewal equation $f_j = g_j + q * f_j$, namely $f_j = f_{g_j}$ in our notations.

However, the function $t \mapsto g_j(i,t)$ is here non-increasing so that we cannot use the bounds given by the first point of Theorem 1 and we have to use the second point, which is more complicated. With that aim, we note that g_j may be written as $g_j = I_j - u_j$ with $I_j(i,t) = \mathbf{1}_{\{i=j\}} = I(i,j)$ and $u_j(i,t) = \mathbf{1}_{\{i=j\}}(1 - \mathrm{e}^{-b_i t})$, both being bounded and non-decreasing with respect of t. We then get from Theorem 1 (point 2):

$$f_{I_j}^{h+} - f_{u_j}^{h} \leqslant f_{g_j} = f_{I_j} - f_{u_j} \leqslant f_{I_j}^{h} - f_{u_j}^{h+} \tag{19}$$

with

$$I_j(i, Nh) = I(i, j)$$
$$u_j(i, Nh) = (I - D_h^N)(i, j).$$

The function f_{I_j} that appears in (19) may be simply interpreted: indeed, f_{I_j} is the single solution of the Markov renewal equation

$$f_{I_j}(i,t) = \mathbf{1}_{\{i=j\}} + \sum_{k\in E} \int_0^t q(i,k,\mathrm{d}u) f_{I_j}(k, t-u)\mathrm{d}u$$

and it corresponds to the Markov renewal function (see [2] or [3]). In other words, $f_{I_j}(i,t)$ represents the mean number of visits to state j on $[0,t]$ for the process starting from state i.

Noting that $I_j(\cdot,(k+1)h) - D_h I_j(\cdot,kh) = (I - D_h)(\cdot, j)$ and $u_j(\cdot,(k+1)h) - D_h u_j(\cdot,kh) = (I - D_h)(\cdot, j)$ are independent on k, we use Algorithm 1 for the computation of $f_{I_j}^{h+}$, $f_{u_j}^{h}$, $f_{I_j}^{h}$ and $f_{u_j}^{h+}$, which provides bounds for $P_t(i,j) = f_j(i,t)$ using (19).

In the special case where $E = \{1,\ldots,m\}$ and A is triangular, and in case we are interested in the evaluation of quantities of the shape $\mathbb{P}_i(X_t \geqslant j)$, we may also use alternative bounds from Proposition 2: setting $v_j(i) = 1_{\{i\geqslant j\}} = \sum_{n=j}^m I(i,n)$ and $w_j(i,t) = v_j(i)\mathrm{e}^{-b_i t}$, we get

$$f_{w_j^{h+}}^{h+}(i,t) \leqslant \mathbb{E}_i(1_{\{X_t\geqslant j\}}) = \mathbb{P}_i(X_t \geqslant j) \leqslant f_{w_j^{h}}^{h}(i,t) \tag{20}$$

with

$$w_j^h(\cdot,kh) = w_j(\cdot,(k+1)h) = \sum_{n=j}^m D_h^{k+1}(\cdot,n)$$
$$w_j^{h+}(\cdot,kh) = w_j(\cdot,kh) = \sum_{n=j}^m D_h^{k}(\cdot,n).$$

As $w_j(\cdot,(k+1)h) - D_h w_j(\cdot,kh)$ is independent on k, we use again Algorithm 1 to compute the bounds provided by (20). Such bounds are designed by "Bounds T" ("T" for triangular case) in the following whereas the general ones are called "Bounds".

Such "bounds T" may be of interest for instance in the case where the states are ranked according to their degradation degree and where the system may only degrade with time, this last assumption corresponding to A upper triangular. In that case, it may indeed be of interest to evaluate the probability that the system is not "too" degraded at time t, which corresponds to quantities of the shape $\mathbb{P}_i(X_t < j)$, or alternatively $\mathbb{P}_i(X_t \geqslant j)$. To evaluate such a probability, the bounds provided by (20) are much more simple and performant than those provided by the summation of (19) using $\sum_{n=j}^m \mathbb{P}_i(X_t = n) = \mathbb{P}_i(X_t \geqslant j)$, as will be numerically observed in Section 6.1 (and as expected).

4.2. Approximations

We now come to different approximations for $P_t(i,j) = f_{g_j}(i,t)$ provided by the present method.

A first approximation consists in taking the middle of the bounds provided by (19), namely, $(f_{I_j}^h - f_{u_j}^{h+} + f_{I_j}^{h+} - f_{u_j}^h)/2$.

Also, taking $g_j(i,t) = \mathbf{1}_{\{i=j\}}\mathrm{e}^{-b_i t}$ as in the previous subsection, we know from Theorem 1 that

$$\lim_{h\to+\infty} f_{g_j}^h(i,t) = \lim_{h\to+\infty} f_{g_j}^{h+}(i,t) = f_{g_j}(i,t) = P_t(i,j).$$

For small h, both $f_{g_j}^h(i,t)$ and $f_{g_j}^{h+}(i,t)$ then constitutes other approximations for $P_t(i,j)$, which are referred to as LGA and UGA for "lower" and "upper geometric approximation" (see Section 2). Noting that $g_j(\cdot,(k+1)h) = D_h \times g_j(\cdot,kh)$, we may use Algorithm 1 for their computation (with some simplification however because we here have $X = 0$ in Algorithm 1).

Also, we have from (17,18):

$$f_{g_j}^{h}(\cdot,Nh) = (B_h^{-1}D_h)^N B_h^{-1} g_j(\cdot,0) = (B_h^{-1}D_h)^N B_h^{-1}(\cdot,j),$$
$$f_{g_j}^{h+}(\cdot,Nh) = C_h^N g_j(\cdot,0) = C_h^N(\cdot,j). \tag{21}$$

Such approximations actually are very near from the well-known Euler's forward approximation (FEA). Indeed, taking $t = Nh$ and remembering that

$$P_t = \mathrm{e}^{NhA} = (\mathrm{e}^{hA})^N,$$

FEA consists in approximating e^{hA} by its first-order term from the following approximation:

$$\mathrm{e}^{hA}(i,j) = \begin{cases} 1 - b_i h + \mathrm{o}(h) & \text{if } i = j, \\ h a_{i,j} + \mathrm{o}(h) & \text{if } i \neq j \end{cases} = (I + hA)(i,j) + \mathrm{o}(h) \tag{22}$$

when h goes to 0^+ (see [12], e.g.).

Actually, in term of the Markov process $(X_t)_{t\geqslant 0}$, it is known that when h goes to 0^+:

$$\mathrm{e}^{hA}(i,i) = P_h(i,i) = \mathbb{P}_i(X_h = i) = \mathbb{P}_i(T_1 > h) + \mathrm{o}(h) = \mathrm{e}^{-b_i h} + \mathrm{o}(h) \tag{23}$$

and, for $i \neq j$:

$$\begin{aligned} \mathrm{e}^{hA}(i,j) &= \mathbb{P}_i(T_1 \leqslant h; X_h = j) = \mathbb{P}_i(T_1 \leqslant h; X_{T_1} = j) + \mathrm{o}(h) = \mathbb{P}_i(T_1 \leqslant h) P_{i,j} + \mathrm{o}(h) \\ &= (1 - \mathrm{e}^{-b_i h}) P_{i,j} + \mathrm{o}(h) = a_{i,j} \frac{1 - \mathrm{e}^{-b_i h}}{b_i} + \mathrm{o}(h) \quad \text{if } b_i \neq 0 \end{aligned} \tag{24}$$

(see [3], e.g.).

Remembering that $C_h = D_h + (I - D_h)P$ and consequently

$$C_h(i,j) = \mathbf{1}_{\{i=j\}} \mathrm{e}^{-hb_i} + (1 - \mathrm{e}^{-hb_i}) P_{i,j}$$

for all $i, j \in E$, Eqs. (23) and (24) may be written as $\mathrm{e}^{hA} = C_h + \mathrm{o}(h)$. Due to (21), UGA $(f_{g_j}^{h+})$ simply consists in taking the first-order term in such an expansion of e^{hA} and in writing $P_t = (\mathrm{e}^{hA})^N$ for $t = Nh$ as for FEA. Due to $I + hA = C_h + \mathrm{o}(h)$, UGA then appears as very near from FEA. However, FEA requires $h \leqslant \frac{1}{\max(b_i)}$ to converge (or $I + hA$ would not be non-negative) while there is no requirement by our method. Beside, $1 - b_i h$ is the first-order approximation for $\mathrm{e}^{-b_i h}$ and, due to the previous interpretation, one might expect UGA to be better than FEA.

As far as LGA is concerned, one might easily check that $B_h^{-1}D_h = \mathrm{e}^{hA} + \mathrm{o}(h)$ when h goes to 0^+ as for UGA. However, it seems that there is no direct interpretation in term of $(X_t)_{t\geqslant 0}$ for $B_h^{-1}D_h$ as there is for C_h.

Two other methods very near from FEA and ours may also be found in the literature:

- the "external uniformization" (see [6,14]) where:

  $$\mathrm{e}^{hA} \sim (I - hA)^{-1}$$

  when h goes to 0^+, with $h < \frac{1}{\rho(A)} \leqslant \frac{1}{2\min b_i}$ and $\rho(A)$ the spectral radius for (A),
- the algorithm provided for semi-Markov processes by Cocozza-Eymard in [4,5], which leads to the following approximation in the Markovian case:

  $$\mathrm{e}^{hA}(i,j) \sim \begin{cases} \frac{1}{1+b_i h} & \text{if } i = j, \\ \frac{a_{i,j} h}{1+b_i h} & \text{if } i \neq j \end{cases}$$

  when h goes to 0^+ (with no requirement for h).

To sum up, we now have six different very near approximations, which have to be compared numerically:

- UGA, which corresponds to $f_{g_j}^{h+}$,
- LGA, which corresponds to $f_{g_j}^{h}$,

- Euler's forward approximation: FEA,
- External uniformization: EUA. (a single LU factorization for $I - hA$ is performed to compute quantities like $(I - hA)^{-1}\mathbf{v}$ for EUA),
- Cocozza-Eymard's approximation: CEA,
- The middle of the bounds provided by (19): MB.

5. A benchmark

In order to perform numerical tests to compare the different methods described in the previous section, we consider the following benchmark: a system is considered with n identical and independent components with respective failure and repair rates λ and μ ($n \geqslant 2$). The state-space is $E = \{0, \ldots, n\}$ where state i corresponds to exactly i failed components. A common cause failure may arrive according to a Poisson process with rate $\alpha > 0$ independently of the inner behavior of the system. When a common cause arrives, each up component is instantaneously affected with probability $p \in [0; 1]$ independently of the other components. Similarly, the system is instantaneously controlled at times which are distributed according to a Poisson process with rate $\beta > 0$ independently of the inner behavior of the system. By those controls, some parameters of the system such as voltage, pressure, temperature,... are re-adjusted and the down components are tried to be put again into operation instantaneously with probability $1 - \gamma$ of success ($\gamma \in [0; 1]$) independently one from each other.

The generator matrix of the system submitted to common cause failures and controls is

$$A = A_1 + A_2$$

with

$$A_1(i,j) = \begin{cases} (n-i)\lambda & \text{if } j = i+1 \quad \text{and} \quad 0 \leqslant i \leqslant n-1 \\ i\mu & \text{if } j = i-1 \quad \text{and} \quad 1 \leqslant i \leqslant n \\ -\sum_{j \neq i} A_1(i,j) & \text{if } j = i \quad \text{and} \quad 0 \leqslant i \leqslant n \end{cases}$$

(the generator of the inner system) and setting $q = 1 - p$:

$$A_2(i,j) = \begin{cases} C_{n-i}^{j-i} p^{j-i} q^{n-j} \alpha & \text{if } 0 \leqslant i \leqslant j-1 \leqslant n-1 \\ C_i^{i-j} (1-\gamma)^{i-j} \gamma^j \beta & \text{if } 0 \leqslant j \leqslant i-1 \leqslant n-1 \\ -\sum_{j \neq i} A_2(i,j) & \text{if } j = i \quad \text{and} \quad 0 \leqslant i \leqslant n \end{cases}$$

(the part of the generator due to common cause failures and controls). For example, taking $n = 5$ (and $m = 6$), we get setting $v = 1 - \gamma$:

$$A = \begin{pmatrix} * & 5\lambda + 5p\alpha q^4 & 10p^2\alpha q^3 & 10p^3\alpha q^2 & 5p^4\alpha q & p^5\alpha \\ \mu + \beta v & * & 4\lambda + 4p\alpha q^3 & 6p^2\alpha q^2 & 4p^3\alpha q & p^4\alpha \\ \beta v^2 & 2\mu + 2\beta\gamma v & * & 3\lambda + 3p\alpha q^2 & 3p^2\alpha q & p^3\alpha \\ \beta v^3 & 3\beta\gamma v^2 & 3\mu + 3\beta\gamma^2 v & * & 2\lambda + 2p\alpha q & p^2\alpha \\ \beta v^4 & 4\beta\gamma v^3 & 6\beta\gamma^2 v^2 & 4\mu + 6\beta\gamma^3 v & * & \lambda + p\alpha \\ \beta v^5 & 5\beta\gamma v^4 & 10\beta\gamma^2 v^3 & 10\beta\gamma^3 v^2 & 5\mu + 5\beta\gamma^4 v & * \end{pmatrix},$$

where * are such that summations on lines are 0.

Taking $p = 0$ and $\gamma = 1$ leads to the usual model for a system with n i.i.d. components and sparse tri-diagonal generator matrix. Taking $0 < p < 1$ and $0 < \gamma < 1$ leads to an entirely full generator matrix. Taking $\mu = 0$ and $\beta = 0$ leads to an upper triangular generator. Also, adjusting α, β, λ and μ may lead to disparate (with stiffness problems) or similar values for the b_i's. This example then allows us to perform our tests in very different cases.

6. Numerical experiments

To perform our tests, we use EXPM1 in Matlab as a reference. This function is based on method 3 from [10,11] and uses a "Padé approximation with scaling and squaring" (from the reference book of Matlab): to compute e^B, it first writes $e^B = \left(e^{\frac{B}{2^n}}\right)^{2^n}$ and chooses n such that $\left\|\frac{A}{2^n}\right\|_2 < 1$. A Padé approximation (of the shape $\left[P\left(\frac{B}{2^n}\right)\right]^{-1} Q\left(\frac{B}{2^n}\right)$ with P and Q polynomial) is then used to compute $e^{\frac{B}{2^n}}$. Successive square elevation then provides the result. This method is known to provide very good results. However, it requires matrix products so that it is not adapted for matrix with big size, which is our aim and indeed, Matlab reveals to be very slow compared to all other methods presented here in case of large n. EXPM1 from Matlab is then convenient to perform numerical tests for n not too big as in the following, but it is not adapted for really large n. Note that the present method may however be employed with much bigger n and that tests have already been done. We first test our bounds and then compare the different approximations.

6.1. Test for the bounds

We provide different examples of computation of bounds for $P_t(i,j)$ or for $P_t(i, j_1 \to j_2) := \sum_{k=j_1}^{j_2} P_t(i,k)$ with $t \in [0, t_{\max}]$, which are compared to the results by Matlab. We envision different cases for the parameters: a sparse case, some full cases with eventual stiffness problems, and finally a triangular case.

Example 7. We consider a sparse transition matrix with

$$n = 200, \quad \alpha = 0, \ \beta = 0, \ \lambda = 10^{-6}, \ \mu = 10^{-4}.$$

We first take

$$t_{\max} = 10^3; \quad N = 20; \ n_0 = 1.$$

This means that the interval of interest $[0, 10^3]$ is divided into 20 parts for the approximations ($h = \frac{10^3}{20} = 50$) and that all values are retained ($N_0 = N$). Bounds for $P_t(100;100)$ are plotted in Fig. 1. We can see in such a figure that the bounds are coherent with Matlab results and that they may be made very tight. We now take

$$t_{\max} = 10^5; \quad N = 10^5; \ n_0 = 2 \times 10^3.$$

This means that $[0, 10^5]$ is divided into 10^5 parts for the approximations ($h = \frac{10^5}{10^5} = 1$) but only one out of 2×10^3 are retained and plotted (namely only $N_0 = \frac{10^5}{2 \times 10^3} = 50$ values are retained). Bounds for $P_t(0; 10 \to 200)$ are plotted in Fig. 2. We can see in such a figure that the accuracy lowers with increasing t, which is due to the method.

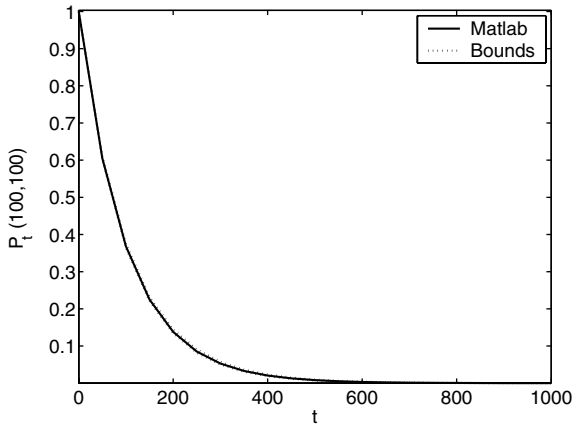

Fig. 1. Example 7, first case. (The bounds and Matlab results are nearly superimposed).

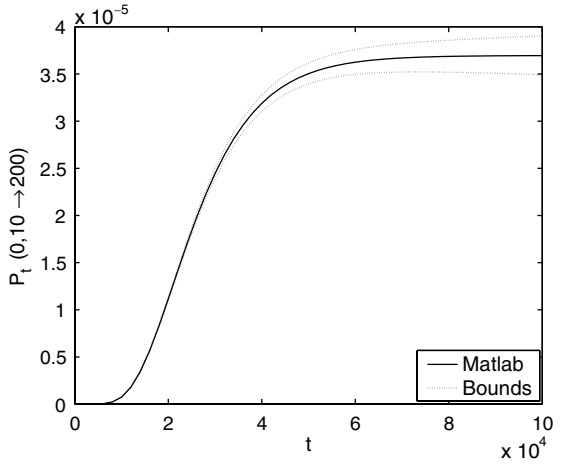

Fig. 2. Example 7, second case.

Example 8. We consider:

$$n = 200; \quad \alpha = 10^{-10}; \ \beta = 10^{-3}; \ p = 0.3; \ \gamma = 0.8, \ \lambda = 10^{-10}, \ \mu = 10^{-4}$$

(not realistic case but full case with stiffness problem due to $\max b_i \simeq 2 \times 10^{-2}$ and $\min b_i \simeq 2 \times 10^{-8}$). We take:

$$t_{\max} = 10^4; \quad N = 10^4; \ n_0 = 50$$

and we compute $P_t(0; 20 \to 200)$ (only $N_0 = 200$ values are retained). The results are displayed in Fig. 3. We can see in such a figure that the method is still valid in case of stiffness problems.

Example 9. We consider

$$n = 200, \quad \alpha = 10^{-6}, \ \beta = 10^{-6}, \ p = 0.5, \ \gamma = 0.5, \ \lambda = 10^{-6}, \ \mu = 10^{-6}$$

(not realistic case but full case with $\max_i b_i \simeq \min_i b_i$). We first take:

$$t_{\max} = 2 \times 10^5; \quad N = 500; \ n_0 = 1$$

and we compute $P_t(150; 150)$. We then take

$$t_{\max} = 3 \times 10^4; \quad N = 5; \ n_0 = 1$$

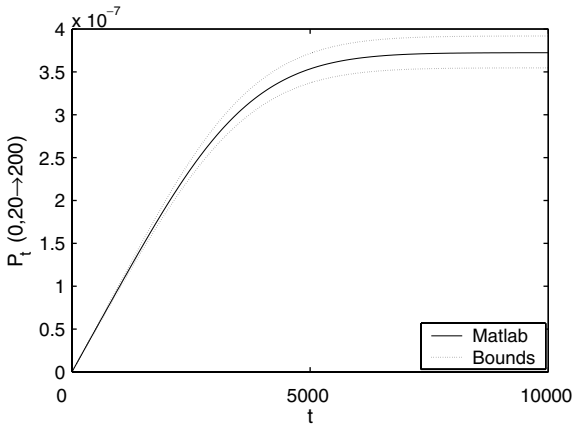

Fig. 3. Example 8.

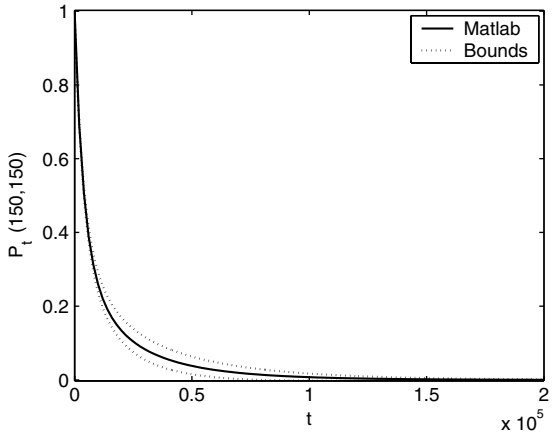

Fig. 4. Example 9, first case.

and we compute $P_t(0;0)$. The results are displayed in Figs. 4 and 5 and are here again coherent with the results from Matlab. The angles in Fig. 5 are due to the fact that the interval of interest has been divided into only 5 parts so that only 5 points have been computed and plotted. This is here enough to get quite accurate values as shows the comparison with Matlab.

Example 10. We take

$$n = 200, \quad \alpha = 10^{-7}, \ \beta = 0, \ p = 0.1, \ \lambda = 10^{-6}, \ \mu = 0$$

(triangular case) and

$$t_{\max} = 1.5 \times 10^5; \quad N = 1500; \ n_0 = 50.$$

Bounds for $P_t(0; 10 \to 200)$ are plotted in Fig. 6 (30 values retained) with the general and the"triangular" methods (see the end of Section 4.1 for details). As expected, the triangular method provides much better results when applicable.

In conclusion of this sub-section, in each case, whatever the parameters are, the bounds are always coherent with the results by Matlab and they can be made as tight as necessary taking the step-size h small enough (or N big enough). As expected, when applicable, the "triangular" method provides much better results than the general one. Also, for a step-size fixed, the tightness of the bounds lowers with increasing t.

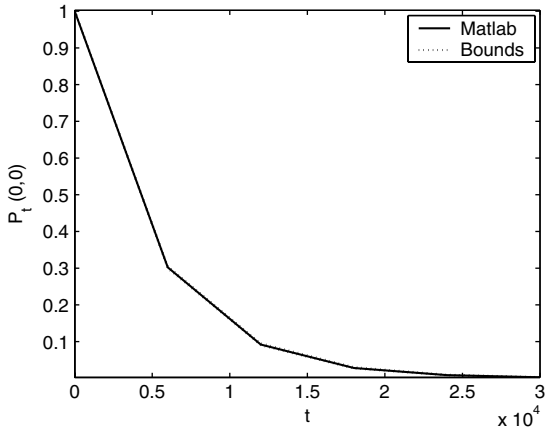

Fig. 5. Example 9, second case. (Only five points plotted. The bounds and Matlab results are nearly superimposed).

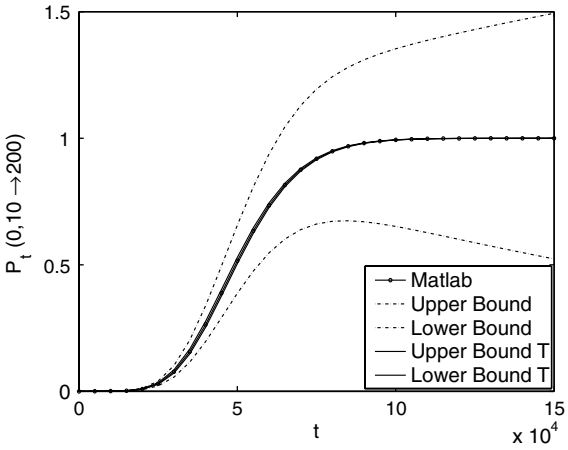

Fig. 6. Example 10, Triangular case. (Bounds T and Matlab results are nearly superimposed).

6.2. Comparison of UGA, LGA and MB

In the following examples, we compare the approximations provided by UGA, LGA and MB.

Example 11. We take

$$n = 20, \quad \alpha = 0, \ \beta = 0, \ \lambda = 10^{-6}, \ \mu = 10^{-5}$$

(sparse case) and

$$t_{\max} = 10^6; \quad N = 2000; \ n_0 = 1.$$

Approximations by UGA, LGA and MB for $P_t(0; 10 \to 20)$ are plotted in Fig. 7. We can see that UGA is the best among the three.

Example 12. We consider

$$n = 20, \quad \alpha = 10^{-6}, \ \beta = 10^{-6}, \ p = 0.5, \ \gamma = 0.5, \ \lambda = 10^{-6}, \ \mu = 10^{-6}$$

(full case). We take

$$t_{\max} = 10^6; \quad N = 666; \ n_0 = 1$$

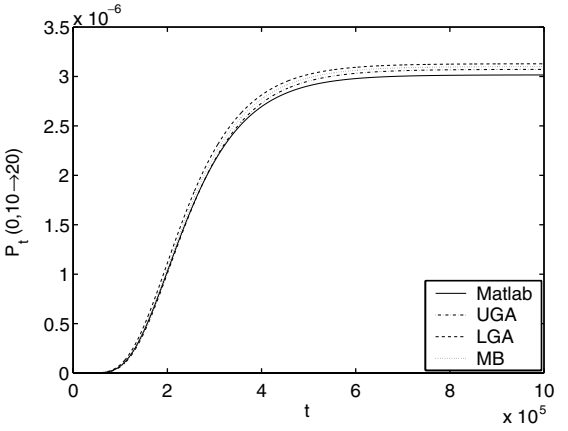

Fig. 7. Example 11.

and approximations by UGA, LGA and MB for $P_t(5;10)$ are plotted in Fig. 8. UGA is here again the best among the three.

Lots of other numerical tests have been performed with bigger n and different values for the parameters (and other benchmarks). The results are similar in each case: for the same value of h, LGA is the looser approximation whereas UGA and MB give roughly similar results, even sometimes a little better for UGA. As MB is longer to compute, UGA is the best approximation among LGA, UGA and MB. Consequently, we only keep UGA in the following.

6.3. Comparison of UGA and CEA

Example 13. We consider

$$n = 20, \quad \alpha = 0, \ \beta = 0, \ \lambda = 10^{-6}, \ \mu = 10^{-4}$$

(sparse case) and we take

$$t_{\max} = 3 \times 10^4; \quad N = 30; \ n_0 = 1.$$

Approximations by UGA and CEA for $P_t(4;4)$ are plotted in Fig. 9. UGA is here better than CEA.

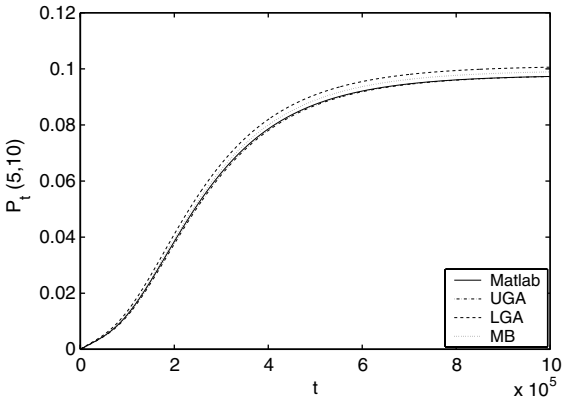

Fig. 8. Example 12.

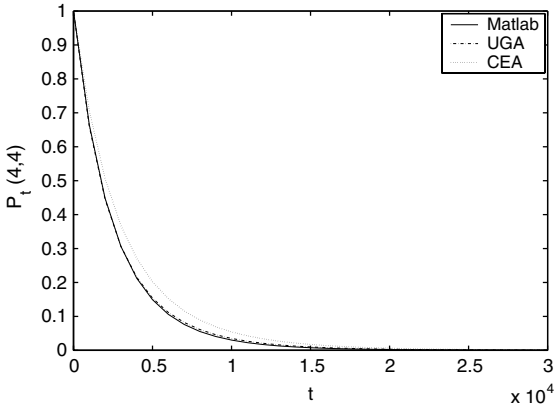

Fig. 9. Example 13.

Example 14. We consider

$$n = 20, \quad \alpha = 10^{-6}, \ \beta = 10^{-6}, \ p = 0.5, \ \gamma = 0.5, \ \lambda = 10^{-6}, \ \mu = 10^{-6}$$

(full case) and we take

$$t_{\max} = 2 \times 10^6; \quad N = 200; \ n_0 = 1.$$

Approximations by UGA and CEA for $P_t(0;5)$ are plotted in Fig. 10. UGA is here better than CEA.

Here again, lots of other numerical tests have been performed with bigger n and different values for the parameters (and other benchmarks). The results are similar in each case: for the same value of h, UGA is better than CEA. Consequently, CEA is not considered any more in the following.

6.4. Comparison of UGA, EUA and FEA

No computation times have been given before because they were very small and comparable, also because the results of comparison between the different methods were quite clear. The comparison now becomes more difficult so that it is worth looking at them.

Example 15. We consider

$$n = 200, \quad \alpha = 0, \ \beta = 0, \ \lambda = 10^{-6}, \ \mu = 10^{-4}$$

(sparse case). In the first case, we take

$$t_{\max} = 10^3; \quad N = 20; \ n_0 = 1.$$

and we plot approximations for $P_t(100;100)$ by UGA, EUA and FEA in Fig. 11. The computations take: 6 c.p.u. by Matlab, 0.01 c.p.u. by FEA and UGA, and 0.04 c.p.u. by EUA (all points for all methods). Here, UGA is better than both EUA and FEA.

In the second case, we take

$$t_{\max} = 10^3; \quad N = 5; \ n_0 = 1$$

and we plot approximations for $P_t(0;10 \to 200)$ by UGA, EUA and FEA in Fig. 12. We can see in such a figure that FEA here requires a smaller h than EUA to converge. Also, UGA is better than EUA (and of course FEA which does not provide correct results).

Example 16. We consider

$$n = 200, \quad \alpha = 10^{-7}, \ \beta = 10^{-6}, \ p = 0.1, \ \gamma = 0.8, \ \lambda = 10^{-6}, \mu = 10^{-5}$$

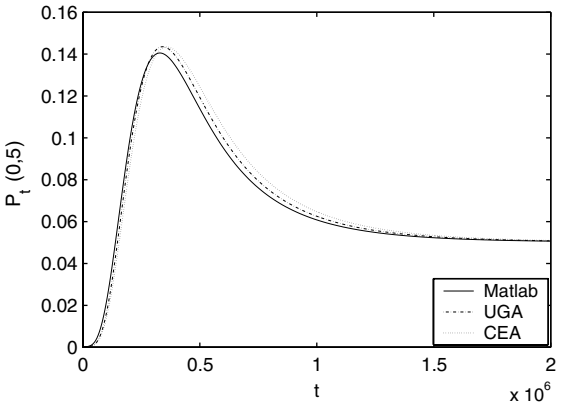

Fig. 10. Example 14.

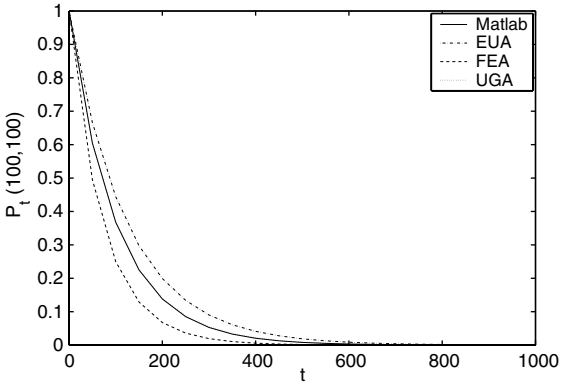

Fig. 11. Example 15, first case. (UGA and Matlab are nearly superimposed).

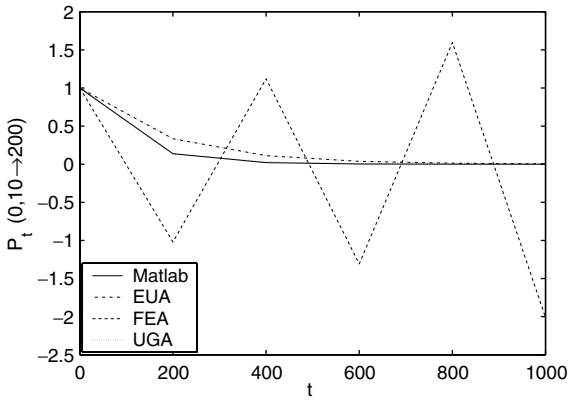

Fig. 12. Example 15, second case. (UGA and Matlab results are nearly superimposed).

(full case) and

$$t_{\max} = 5 \times 10^4; \quad N = 50; \quad n_0 = 1.$$

We plot approximations for $P_t(0; 0 \to 2)$ by UGA, EUA and FEA in Fig. 13. The computations take: 8 c.p.u. by Matlab, 0.03 c.p.u. by FEA, 0.08 c.p.u. by UGA, and 0.3 c.p.u. by EUA. Here, both EUA and FEA are slightly better than UGA.

Example 17. We consider

$$n = 200, \quad \alpha = 10^{-6}, \quad \beta = 10^{-6}, \quad p = 0.5, \quad \gamma = 0.5, \quad \lambda = 10^{-6}, \quad \mu = 10^{-6}$$

(full case). In the first case, we take

$$t_{\max} = 2 \times 10^5; \quad N = 50; \quad n_0 = 1$$

and we plot approximations for $P_t(150; 150)$ by UGA, EUA and FEA in Fig. 14. In this case, we have $h < \frac{1}{\max_i b_i} = 5 \times 10^3$. However FEA does not provide correct results. UGA is better than EUA for small t, the contrary is true for large t. The computations take: 7 c.p.u. by Matlab, 0.031 c.p.u. by FEA, 0.06 c.p.u. by UGA, and 0.27 c.p.u. by EUA.

In the second case, we take

$$t_{\max} = 2 \times 10^4; \quad N = 20; \quad n_0 = 1$$

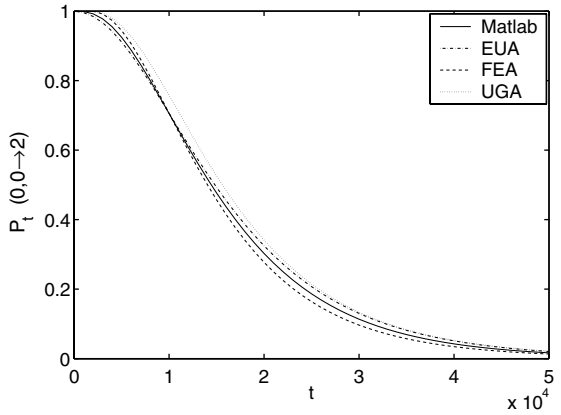

Fig. 13. Example 16.

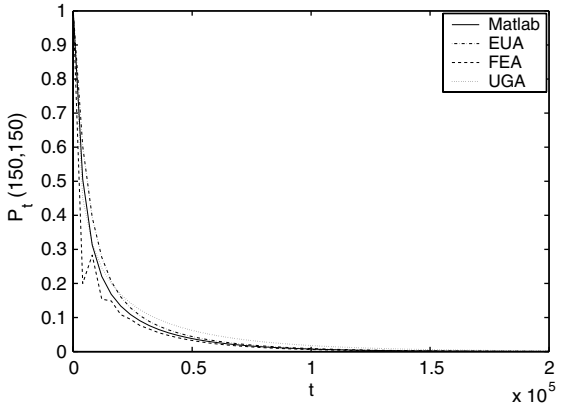

Fig. 14. Example 17, first case.

and we plot approximations for $P_t(0;0)$ by UGA, EUA and FEA in Fig. 15. Here, UGA is better than both EUA and FEA.

Lots of other tests have been performed which are not provided here. As for c.p.u. times, FEA and UGA (with the same h) are very similar with FEA slightly quicker, EUA is a little longer (from twice up to ten times longer in our tests), whereas Matlab is clearly much longer. As for the performance, most of the time, UGA is

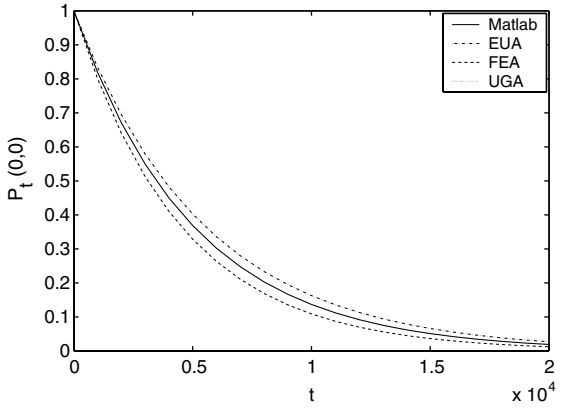

Fig. 15. Example 17, second case. (UGA and Matlab results are nearly superimposed).

better, especially for t not too big. However, FEA and/or EUA are sometimes slightly better, in particular for large t, when approaching the long-time run. Note that UGA has not been compared here to FEA and EUA in the special case of an triangular generator with interest in quantities like $\mathbb{P}_i(X_t \geqslant j)$, which would clearly be to the advantage of UGA.

7. Conclusion

We have proposed, in this paper, bounds and approximations for time-dependent quantities for big Markovian systems, which are solutions of Markov renewal equations. We have focused on the transition probabilities which have been the object of an extensive literature. The quality of the bounds has been tested on different numerical examples showing the accuracy of the method. Also, the bounds and approximations have been interpreted in term of the Markov process showing the proximity of our method with standard methods such as Euler's forward approximation (FEA) or external uniformization (EUA), and with the method from [4] (CEA). Different numerical tests have then been performed for comparison purpose: first, our different approximations have been compared one with each other and one of them has emerged, the so-called UGA for Upper Geometrical Approximation. Then, UGA has been tested versus CEA with a clear advantage to UGA. Finally, UGA has been tested versus FEA and EUA with general advantage to UGA but exceptions however, especially in long time run.

References

[1] P.G. Ciarlet, Introduction to Numerical Linear Algebra and Optimisation, Cambridge University Press, 1989.
[2] E. Cinlar, Introduction to Stochastic Processes, Prentice-Hall, Englewood Cliffs, NJ, 1975.
[3] C. Cocozza-Thivent, Processus stochastiques et fiabilité des systèmesMathématiques et Applications, vol. 28, Springer, 1997 (in French).
[4] C. Cocozza-Thivent, R. Eymard, Approximation of the marginal distributions of a semi-Markov process using a finite volume scheme, ESAIM: M2AN 38 (5) (2004) 853–875.
[5] C. Cocozza-Thivent, R. Eymard, Numerical computation of the marginal distributions of a semi-Markov process, in: Pham, H. (Ed.), Reliability Modelling, Analysis and Optimization, in press.
[6] M. Kijima, Markov Processes for Stochastic modelling, Chapman & Hall, London, 1997.
[7] S. Mercier, Encadrement de quantités fiabilistes pour un gros système markovien, Actes du congrès Lambda-Mu 15 (15e Congrès de Maîtrise des Risques et de Sûreté de Fonctionnement), Lille, octobre 2006 (in French).
[8] S. Mercier, Discrete random bounds for general random variables and applications to reliability, European Journal of Operational Research 177 (1) (2007) 378–405, doi:10.1016/j.ejor.2005.09.043 (available online 15 Feb. 2006).
[9] S. Mercier, Numerical bounds for semi-markovian quantities and applications to reliability, Methodology and Computing in Applied Probability (in revision).
[10] C.B. Moler, C.F. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, SIAM Review 20 (1979) 801–836.
[11] C.B. Moler, C.F. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Review 45 (2003) 3–49.
[12] A. Rauzy, An experimental study on iterative methods to compute transient solutions of large Markov models, Reliability Engineering & System Safety 86 (1) (2004) 105–115.
[13] A. Reibman, R.M. Smith, K.S. Trivedi, Markov and Markov reward models: A survey of numerical approaches, European Journal of Operations Research 40 (1989) 257–267.
[14] S.M. Ross, Introduction to Probability Models, eighth ed., Academic Press, Amsterdam, 2003.
[15] R.B. Sidje, W.J. Stewart, A numerical study of large sparse matrix exponentials arising in Markov chains, Computational Statistics & Data Analysis 29 (3) (1999) 345–368.
[16] W.J. Stewart, Introduction to the Numerical Solution of Markov Chains, [B], Princeton Univ. Press, Princeton, NJ, 1994.