Transient Analysis of CTMCs: Uniformization or Matrix Exponential?

Reza Pulungan and Holger Hermanns

***Abstract*—Many numerical solution methods exist for the transient analysis of continuous-time Markov chains, such as differential equations, matrix exponential and uniformization methods. Uniformization is preferable for reasons of its numerical stability, pre-determined error bounds and acceptable time complexity. However, latest results reveal that computation of matrix exponentials can be carried out much more efficiently than before. In this paper, we compare uniformization and matrix exponential—through Padé approximation—coupled with the scaling and squaring method tailored to the IEEE floating-point specification. We show that, in certain circumstances, the use of Padé approximation is advisable, even though in most cases uniformization performs better.**

***Index Terms*—Transient analysis, continuous-time Markov chains, uniformization, matrix exponential, Padé approximation, scaling and squaring.**

I. Introduction

MARKOV chains form a class of stochastic processes that satisfy Markov property. This property is important for it renders the analysis of these processes tractable and furthermore suitable for various modelling techniques. This paper deals with discrete-state, continuous-time Markov processes: continuous-time Markov chains (CTMCs). CTMCs have been intensively studied and many textbooks ([1], [2], [3], [4]) provide thorough introduction to this field of research. CTMCs have also been used in diverse areas of applications, including reliability [5], forecasting [6], transportation [7], and even pre-clinical study [8].

Analysis of CTMCs is usually conducted by computing transient or steady-state probabilities. The computation of steady-state probabilities can be reduced to the solution of a system of linear equations. Transient-state probabilities, on the other hand, can be computed by solving a system of Kolmogorov differential equations. The solution of the system of differential equations can further be carried out directly or through an exponential function. In 1953, Jensen [9] introduced a new method to compute these transient-state probabilities, which he called randomization. This method, later called uniformization, avoids direct exponentiation, but instead evaluates a truncated formal power series of the exponential function. Direct exponentiation or its approximation is avoided because of its poor numerical properties and severe round-off errors [10]. Further investigations and improvements on the application of uniformization in transient analysis of CTMCs can be found in [11], [12], [10], [13], [14], [15].

As previously mentioned, transient-state probabilities can also be computed through an exponential function of matrices. In this area, Moler and Van Load [16] presented a survey of the various methods for computing matrix exponentials, which was later revisited and refined in [17]. Researches in this field have flourished, resulting in more efficient ways of computing matrix exponentials. Sidje presented a tool `Expokit` in [18], a matrix exponential package that can deal with many types of matrices. Higham [19] sharpened previous relative perturbation errors and proposed a faster method for computing matrix exponentials.

With such advancements in matrix exponential methods, we are interested in finding out whether it is still wise to use uniformization to perform transient analysis in CTMCs and whether we should consider using matrix exponential instead. To that end, we compare uniformization and matrix exponential, through Padé approximation combined with scaling and squaring. As a result of our study, we identify circumstances, under which matrix exponential is more effective than uniformization, and then discuss the limitations of these observations.

The rest of the paper is organized as follows: Section II presents the background of transient analysis of CTMCs. The two methods, uniformization and matrix exponential, are described in Section III. In Section IV, we analyze both methods by comparing them in terms of time complexity and storage requirements. We also present the results of experiments to compare the performance of their implementations. Section V concludes the paper.

II. Transient Analysis of CTMCs

Let state space $\mathcal{S}$ be finite and discrete, and set $\mathcal{T}$ be continuous. A CTMC is a stochastic process $\{X(t) \in \mathcal{S} \mid t \in \mathcal{T}\}$ satisfying Markov property:

$$\begin{aligned}\Pr\{X(t_{n+1}) = s_{n+1} \mid X(t_0) = s_0, \ldots, X(t_n) = s_n\} \\ = \Pr\{X(t_{n+1}) = s_{n+1} \mid X(t_n) = s_n\}.\end{aligned}$$

With every pair of states $s, s' \in \mathcal{S}$, we associate a *rate* $\mathbf{R}(s, s')$ that determines the delay and the probability of the transition from s to s'. The rates of the transitions between all pairs of states, hence, form a matrix $\mathbf{R}$. Let $E(s) = \sum_{s' \in \mathcal{S}} \mathbf{R}(s, s')$, namely the total rate of taking an outgoing transition from state s. Markov property can only be satisfied if the sojourn time of each state in the CTMC is governed by a memoryless distribution. Since only *negative exponential distributions* are memoryless, these state sojourn

Manuscript received August 23, 2017; revised December 7, 2017. This work has been supported by the European Research Council through Advanced Investigators Grant 695614 (POWVER) and by the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, grant no. 97/J01.1.28/PL.06.02/2013.

R. Pulungan is with the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Yogyakarta, Indonesia, e-mail: pulungan@ugm.ac.id. He is the corresponding author.

H. Hermanns is with the Dependable Systems and Software group, Department of Computer Science, Saarland University, Saarbrücken, Germany, e-mail: hermanns@cs.uni-saarland.de.

times are negative exponentially distributed. The sojourn-time distribution of state s is given by:

$$F_s(t) = 1 - \mathrm{e}^{-E(s)t}, \quad t \geq 0.$$

For many measures over CTMCs, it is interesting to know the probability that a CTMC is *residing in* a particular state after a given time interval has elapsed. This probability is called the *transient-state probability*. Let $\vec{\pi}(t)$ be the probability vector of these transient-state probabilities (for all states) at time instant t; then it satisfies the system of Kolmogorov differential equations:

$$\frac{d}{dt}\vec{\pi}(t) = \vec{\pi}(t)\mathbf{Q}, \quad \vec{\pi}(0) = \vec{\pi}_0, \tag{1}$$

where:

$$\mathbf{Q} = \mathbf{R} - Diag(E),$$

is the *infinitesimal generator matrix*, $Diag(E)$ is a diagonal matrix formed by $E(s)$'s, and $\vec{\pi}_0$ is an initial probability vector. The solution of Equation (1) is given by:

$$\vec{\pi}(t) = \vec{\pi}_0 \mathrm{e}^{\mathbf{Q}t} = \vec{\pi}_0 \sum_{i=0}^{\infty} \frac{(\mathbf{Q}t)^i}{i!}. \tag{2}$$

III. Two Methods for Transient Analysis

In this section, the two methods for computing the transient-state probabilities of CTMCs are described.

A. Uniformization

Uniformization is one of the methods used to perform transient analysis of CTMCs. In uniformization, a CTMC is first uniformized and the resulting embedded discrete-time Markov chain (DTMC) is then analyzed. To uniformize a CTMC, which is specified by initial probability distribution $\vec{\pi}_0$ and infinitesimal generator matrix $\mathbf{Q}$, a rate of sojourn Λ is selected such that $\Lambda \geq \max\{E(s)\}$. This rate is then set to become the rate of all states, hence the name uniformization. The probability matrix of the embedded DTMC is then:

$$\mathbf{P} = \mathbf{I} + \frac{\mathbf{Q}}{\Lambda},$$

where $\mathbf{I}$ is the corresponding identity matrix. Once such a DTMC is available, $\vec{\pi}(t)$, the transient-state probabilities vector at time t in Equation (2), can be rewritten as:

$$\vec{\pi}(t) = \vec{\pi}_0 \mathrm{e}^{\Lambda(\mathbf{P}-\mathbf{I})t} = \vec{\pi}_0 \sum_{i=0}^{\infty} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^i}{i!} \mathbf{P}^i. \tag{3}$$

The expression $\mathrm{e}^{-\Lambda t}\frac{(\Lambda t)^i}{i!}$ is the distribution of a Poisson process $\{N_t, t \geq 0\}$ with rate Λ. In practice, the infinite sum in Equation (3) is truncated at some depth, say N, hence:

$$\vec{\pi}(t) \approx \vec{\pi}_0 \sum_{i=0}^{N} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^i}{i!} \mathbf{P}^i. \tag{4}$$

Given ϵ, the maximum error allowed for any component of the transient-state probabilities vector, N is the smallest integer such that:

$$1 - \sum_{i=0}^{N} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^i}{i!} \leq \epsilon.$$

However, when the value of Λt gets bigger, the Poisson distribution gets thinner. In this case, the Poisson distribution for small i's will be negligibly small. Avoiding the summation in Equation (4) for such small i's will not only reduce computational overhead considerably, but also evade underflow and overflow problems. Fox and Glynn in [11] observed this and proposed an algorithm to perform truncation on both sides of the summation. Their algorithm also provides a way to obtain the values of Poisson probabilities without evaluating expression $\mathrm{e}^{-\Lambda t}$ directly. Let L and U be the lower and upper bounds of the number of terms to compute in the summation, respectively. Given ϵ, then L and U are the largest and smallest integers, respectively, such that:

$$\sum_{i=0}^{L} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^i}{i!} \leq \frac{\epsilon}{2}, \quad \text{and} \quad 1 - \sum_{i=0}^{U} \mathrm{e}^{-\Lambda t} \frac{(\Lambda t)^i}{i!} \leq \frac{\epsilon}{2}. \tag{5}$$

Algorithm 1 computes the transient-state probabilities vector by using uniformization. Function foxglynn() is the Fox and Glynn's algorithm. Algorithm 1 returns L, the lower truncation point; U, the upper truncation point; a vector $\vec{w}$ and a value W such that $\vec{w}[i]/W$ is the probability that i events occur in the Poisson process.

Algorithm 1: Transient analysis by uniformization

Input: $\vec{\pi}_0, \mathbf{Q}, t, \epsilon$
Output: $\vec{\pi}$

```
begin
    Λ ⟵ max{E(s)}
    P ⟵ Q/Λ + I
    [L, U, w, W] ⟵ foxglynn(Λt, ε)
    p ⟵ π0 P^(L-1) w[L-1]/W
    for L to U do
        π ⟵ π + p
        p ⟵ pP w[i]/W
    end
    return π
end
```

B. Matrix exponential

Many methods for computing matrix exponentials of the form of Equation (2) exist. Moler and Van Loan [16], [17] presented a thorough survey for such methods. We are particularly interested in Padé approximation method combined with the scaling and squaring method. Padé approximation is a method for approximating a formal power series with a rational function. This method is attributed to Henri Eugène Padé (1863-1953). The rational function is derived by expanding the power series as a ratio of two polynomials and then determining the coefficients of the numerator and denominator polynomials.

Consider the following arbitrary formal power series:

$$f(t) = \sum_{i=0}^{\infty} c_i t^i, \quad c_i \in \mathbb{R}. \tag{6}$$

The power series can be approximated by a rational function:

$$R_{m,n}(t) = \frac{P_{m,n}(t)}{Q_{m,n}(t)},$$

where $P_{m,n}(t)$ and $Q_{m,n}(t)$ are two polynomials of degree m and n, respectively. Such rational function is called a *Padé approximant* of the power series if the following conditions:

$$R_{m,n}(0) = f(0) \quad \text{and} \quad \frac{d^k}{dt^k} R_{m,n}(t)\Big|_{t=0} = \frac{d^k}{dt^k} f(t)\Big|_{t=0}$$

hold for $k = 1, 2, \ldots, m+n$.

The magnitude of the approximation error for a particular Padé approximant is given in Theorem 1.

Theorem 1 ([20]): Let $P_{m,n}(t)$ and $Q_{m,n}(t)$ be polynomials of degree m and n, respectively, and let $f(t)$ be a formal power series, then:

$$\frac{P_{m,n}(t)}{Q_{m,n}(t)} - f(t) = \mathcal{O}(t^{m+n+1}).$$

Conversely, let $W(t)$ be a polynomial of degree m and $V(t)$ be a polynomial of degree n such that:

$$\frac{W(t)}{V(t)} - f(t) = \mathcal{O}(t^{m+n+1}),$$

then:

$$\frac{W(t)}{V(t)} = \frac{P_{m,n}(t)}{Q_{m,n}(t)}.$$

Theorem 1 establishes that there exists a unique Padé approximant of particular degrees of numerator and denominator polynomials for any power series; and for such an approximant a larger t results in a bigger approximation error. Brezinski [20] presented a constructive method to build Padé approximants for arbitrary power series. Diagonal Padé approximants $R_{m,m}(\cdot)$ are more preferable than off-diagonal approximants, since for the same order of approximation, diagonal approximants expend the least amount of work. For the special case when the power series is an exponential function, the generic forms of the two polynomials $P_{m,n}(t)$ and $Q_{m,n}(t)$ have been determined [20], namely:

$$P_{m,n}(t) = \sum_{i=0}^{m} \frac{(m+n-i)!m!}{(m+n)!(m-i)!} \frac{t^i}{i!}, \quad \text{and}$$

$$Q_{m,n}(t) = \sum_{i=0}^{n} \frac{(m+n-i)!n!}{(m+n)!(n-i)!} \frac{(-t)^i}{i!}.$$

Moler and Van Loan described in [17] that Padé approximation coupled with the scaling and squaring technique, when properly implemented, is one of the most effective methods to compute matrix exponentials. Scaling and squaring method exploits an important property of exponential functions, namely:

$$\mathrm{e}^t = (\mathrm{e}^{t/m})^m.$$

In computing matrix exponential $\mathrm{e}^{\mathbf{A}}$, for the same order of approximation error, polynomials $P_{m,n}(\mathbf{A})$ and $Q_{m,n}(\mathbf{A})$ of higher degrees are required if the norm of the matrix ($\|\mathbf{A}\|$) increases. The scaling and squaring method enables us to choose s, for which $\mathrm{e}^{\mathbf{A}/2^s}$ can be computed efficiently with reasonable degrees of $P_{m,n}(\mathbf{A})$ and $Q_{m,n}(\mathbf{A})$ (since $\|\mathbf{A}/2^s\|$ stays small) and then by squaring the result s times, to obtain the final result $(\mathrm{e}^{\mathbf{A}/2^s})^{2^s} = \mathrm{e}^{\mathbf{A}}$. A common criterion is usually to select s such that $\|\mathbf{A}/2^s\| \leq \frac{1}{2}$.

When the scaled matrix $\mathrm{e}^{\mathbf{A}/2^s}$ is approximated by $R_{m,m}(\mathbf{A}/2^s)$, naturally an approximation error occurs. If:

$$R_{m,m}\left(\mathbf{A}/2^s\right)^{2^s} = \mathrm{e}^{\mathbf{A}+\mathbf{E}},$$

where $\mathbf{E}$ is the perturbation matrix, then it is shown in [17] that:

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq 8 \left(\frac{\|\mathbf{A}\|}{2^s}\right)^{2m} \frac{(m!)^2}{(2m)!(2m+1)!}. \tag{7}$$

Given an upper bound ϵ, the values of s and m can be chosen such that:

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \epsilon.$$

If this is the case, it can be shown that [17]:

$$\frac{\|R_{m,m}(\mathbf{A}/2^s)^{2^s} - \mathrm{e}^{\mathbf{A}}\|}{\|\mathrm{e}^{\mathbf{A}}\|} \leq \epsilon \|\mathbf{A}\| \mathrm{e}^{\epsilon\|\mathbf{A}\|}, \tag{8}$$

which gives the relative error bound of the approximant to its exact value.

It must be remembered that the above-mentioned error bounds are due to truncation error when computing the infinite sums of the power series. The whole derivation is carried out with the assumption of exact arithmetic. The problem of rounding error, however, is important, especially when the squaring is performed more often (namely when s is large). It is worth noting that function `expm` in MATLAB [21], which computes matrix exponential, uses this result with $m = 6$ and s such that $\|\mathbf{A}/2^s\|_\infty < \frac{1}{2}$.

The relative perturbation error in Equation (7) is sharpened by Higham in [19] as described in Theorem 2.

Theorem 2 ([19]): Let the diagonal Padé approximant $R_{m,m}(\cdot)$ satisfy:

$$\frac{R_{m,m}(\mathbf{A}/2^s)}{\mathrm{e}^{\mathbf{A}/2^s}} = \mathbf{I} + \mathbf{G},$$

where $\|\mathbf{G}\| < 1$. Then:

$$R_{m,m}(\mathbf{A}/2^s)^{2^s} = \mathrm{e}^{\mathbf{A}+\mathbf{E}},$$

where $\mathbf{E}$ commutes with $\mathbf{A}$ and the relative error:

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \frac{-\log(1-\|\mathbf{G}\|)}{\|\mathbf{A}/2^s\|}.$$

We can then write the relative error in Theorem 2 as:

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \frac{-\log(1-g(\theta))}{\theta}, \tag{9}$$

with $\theta = \|\mathbf{A}/2^s\|$ and:

$$g(\theta) = \sum_{i=2m+1}^{\infty} |c_i| \theta^i.$$

Higham in [19] evaluated $g(\theta)$ for $m = 1, \ldots, 21$ such that the relative error in Equation (9) does not exceed the unit round-off in IEEE double-precision arithmetic, namely $u = 2^{-53} \approx 1.1 \times 10^{-16}$. Choosing $m = 13$—which gives $\theta = 5.371920351148152$—is the most preferable for it requires the lowest number of matrix multiplications in the evaluation of the Padé approximant and the squaring. Note that using this method, $\|\mathbf{A}/2^s\| \leq \theta$ is very big compared to the criterion proposed in [17] ($\|\mathbf{A}/2^s\| < \frac{1}{2}$), which means the number of matrix multiplications required in the squaring phase is even smaller.

Furthermore, an efficient scheme can be applied in the evaluation of the numerator and the nominator of $R_{m,m}(\mathbf{A})$. Since $Q_{m,m}(\mathbf{A}) = P_{m,m}(-\mathbf{A})$, only even powers of $\mathbf{A}$ are required to be computed. For instance, for $m = 13$, let:

$$U = \mathbf{A}(\mathbf{A}^6(b_{13}\mathbf{A}^6 + b_{11}\mathbf{A}^4 + b_9\mathbf{A}^2) + b_7\mathbf{A}^6 + b_5\mathbf{A}^4 + b_3\mathbf{A}^2 + b_1\mathbf{I})$$

and:

$$V = \mathbf{A}^6(b_{12}\mathbf{A}^6+b_{10}\mathbf{A}^4+b_8\mathbf{A}^2)+b_6\mathbf{A}^6+b_4\mathbf{A}^4+b_2\mathbf{A}^2+b_0\mathbf{I},$$

then $P_{13,13}(\mathbf{A}) = U + V$ and $Q_{13,13}(\mathbf{A}) = -U + V$. Therefore, the evaluation of both $P_{13,13}(\mathbf{A})$ and $Q_{13,13}(\mathbf{A})$ requires only 6 matrix multiplications. Nevertheless, it must not be forgotten that $R_{m,m}(\mathbf{A})$ itself is computed by solving the system of linear equations $Q_{m,m}(\mathbf{A})R_{m,m}(\mathbf{A}) = P_{m,m}(\mathbf{A})$.

For $\|\mathbf{A}\|_1 > 1$, Higham's method requires two or three fewer matrix multiplications than `expm`, and four to five fewer than `padm`. `padm` is the Padé approximation to matrix exponential in the tool `Expokit` ([18]). For $\|\mathbf{A}\|_1 \leq 1$, Higham's method requires up to (no more than) 3 and 5 fewer matrix multiplications, compared to `expm` and `padm`, respectively.

IV. Methods Comparison

In this section, we compare uniformization and matrix exponential methods. First, the worst-case computational requirements of both methods are analyzed. Several practical issues concerning the complexity and implementation of both methods are also considered. The next two subsections describe the experiments conducted to compare the performance of both methods. The section is closed by a general discussion of the comparison.

A. Worst-case complexity

Let $\vec{\pi}_0$ and $\mathbf{Q}$ be the initial probability distribution and the infinitesimal generator matrix of a CTMC, respectively. Transient analysis of the CTMC by uniformization proceeds by applying Algorithm 1. In this case, it is advisable to choose Λ, the uniformizing rate, to be strictly larger than the largest absolute diagonal element of $\mathbf{Q}$ to ensure that the embedded DTMC be aperiodic and hence has a steady-state solution. On the other hand, transient analysis of the CTMC by matrix exponential method is carried out by evaluating Equation (2). For a particular time-instant t, the multiplication of matrix $\mathbf{Q}$ with constant t is calculated and then expression $\mathrm{e}^{\mathbf{Q}t}$ is evaluated by using matrix exponential with Padé approximant $R_{13,13}(\mathbf{Q}t)$ and scaling and squaring method as required by the norm $\|\mathbf{Q}t\|$. If $\|\mathbf{Q}t\| > \theta$ then the matrix is scaled s times such that $\|\mathbf{Q}t/2^s\| \leq \theta$, where $\theta = 5.371920351148152$.

Let us consider the complexity of transient analysis by matrix exponential when the CTMC has N states. We assume that a standard matrix multiplication requires a cubic number of element-wise multiplications, and do not consider specialized algorithms such as [22]. The number of multiplications required to compute matrix exponential of the scaled matrix $(\mathbf{Q}t/2^s)$ is $6N^3$, since using Padé approximant $R_{13,13}(\cdot)$ requires 6 matrix multiplications. Furthermore, the resulting matrix is squared s times to produce the final matrix exponential and this requires sN^3 multiplications, where $s = \lceil \log_2(\frac{\|\mathbf{Q}t\|}{\theta}) \rceil$. The final vector-matrix multiplication of the initial probability distribution with the result of matrix exponential adds N^2 multiplications. Thus, overall, the number of multiplications required when using ∞-norm is:

$$\begin{aligned}\mathcal{O}(N^3(6+s)+N^2) &= \mathcal{O}(N^3(6+\lceil \log_2(\frac{\|\mathbf{Q}t\|_\infty}{\theta})\rceil)+N^2),\\ &= \mathcal{O}(N^3(6+\lceil \log(\Lambda t)-\log(\theta)\rceil)+N^2),\end{aligned}$$

where $\Lambda = \max\{E(s)\}$. This reduces to $\mathcal{O}(N^3(\lceil \log(\Lambda t)\rceil))$, since θ is a constant.

In transient analysis via uniformization, the number of terms needed in the summation in Equation (3), if truncated from both sides, is in the order of $\sqrt{\Lambda t}$ [11]. Since each term requires a vector-matrix multiplication, computing the final transient-state probabilities vector after the left truncation requires $\mathcal{O}(N^2\sqrt{\Lambda t})$ multiplications. However, the left truncation removes the first $\mathcal{O}(\Lambda t)$ terms from the sum in Equation (3). To obtain the first significant term after the left truncation requires $\mathcal{O}(\Lambda t)$ vector-matrix multiplications, thus $\mathcal{O}(N^2\Lambda t)$ multiplications. Overall, the number of multiplications required in uniformization is in the order of $(N^2\Lambda t + N^2\sqrt{\Lambda t})$, hence $\mathcal{O}(N^2\Lambda t)$.

When N is large compared to Λt (which is very common), then uniformization is the method of choice. However, when Λt is large compared to the state-space size N, the $\mathcal{O}(\Lambda t)$ vector-matrix multiplications needed to reach the left truncation point constitute a considerable overhead. This procedure can better be replaced by successive squaring of $\mathbf{P}$, the transition probabilities matrix of the embedded DTMC, which requires $\log(\Lambda t)$ matrix multiplications. In this fashion, the number of multiplications required in uniformization is in the order of $(N^3\log(\Lambda t) + N^2\sqrt{\Lambda t})$, hence $\mathcal{O}(N^3\log(\Lambda t))$.

The previous analysis reveals rough parameter ranges where the computational requirements for uniformization are expected to be lower than those for matrix exponential, namely when $N > \frac{\Lambda t}{\log(\Lambda t)}$. It also suggests that, theoretically, we can expect matrix exponential to perform better when Λt is large enough, namely when $N < \frac{\Lambda t}{\log(\Lambda t)}$.

B. Practical considerations

In the previous subsection, we have shown that matrix exponential may, theoretically, perform better if Λt is large enough. The expression Λt represents the *index of stiffness* of the model. The index of stiffness reflects the number of occurrences of the most frequent event during the observation time. For ordinary models, this index is expected to be small, since most of the interesting properties can be observed in comparatively small time-intervals (small time-intervals for large rates and vice versa). On the other hand, if the ratio of the largest to the smallest rates is large, and we are more interested in the occurrences of those events that happen with small rates, then we are forced to observe the model in large time-intervals.

In practice, very stiff models are not rare at all [23]: large values for Λt are notorious for reliability models and their evaluation. In reliability evaluation, within a model, the rate of occurrences of normal events are usually much larger than the rate of occurrences of events that lead to failures. Since in reliability evaluation, we are mostly interested in events that lead to failures, the model must be analyzed at large time-instant t's. At the same time, reliability models of small and medium sizes (less than a few thousand states) are rather common [24], [25], [26]. Thus, reliability models are prime candidates where matrix exponential tailored to the IEEE floating-point specification might show its advantages.

Each of uniformization and matrix exponential methods provides a way to compute an error bound. The error bound in uniformization is the maximum deviation of each component of the resulting transient-state probabilities vector.

This bound is pre-determined by setting ϵ in Equation (5). For matrix exponential, the error bound is the maximum perturbation error of the approximation relative to its exact value as given by Equation (8). With ϵ in Equation (8) being fixed to the unit round-off of IEEE double-precision arithmetic, the error bound of this method is comparable to having ϵ in Equation (5) set to 10^{-10} up to 10^{-12}.

C. Experimental evaluation

For models with low index of stiffness, previous researches indicate that uniformization is the method of choice amongst all existing methods. In this subsection, we compare the performance of uniformization and matrix exponential methods experimentally for stiff models. For that purpose, we have selected two CTMC models, namely a tandem queueing network and a cyclic server polling system models. Both models are stiff, and, thus, help to show the applicability of matrix exponential method.

Experiment 1: Simple Tandem Queueing Network [27]. This is a CTMC model of $M/Cox_p/1$ queue of a certain capacity c. Arrival to the queue is exponentially distributed while its service time is distributed according to phase-type distribution. Cox_p represents the phase-type distribution of Coxian representation with p phases. In this experiment, we use queue with phases $p = 2$.

The computation times of both methods are compared for various cases by varying the capacity of the queues (which corresponds to the number of states in the model, ranging from 45 to 861 states) and the times at which the transient-state probabilities are inspected (ranging from 1 to 10,000). The maximum of absolute diagonal elements of the generator matrices, namely $\max\{E(s)\}$, of all models ranges from 22 for the model with 45 states to 86 for the model with 861 states.

Experiment 2: Cyclic Server Polling System [28]. A polling system consists of several queueing stations and a polling server. The server serves each queue in a cyclic manner. In this experiment, the server only has a single buffer and must handle 4 to 7 queueing stations.

The computation times of both methods are compared for the cases where the number of stations handled by the polling server is varied from 4 to 7. The resulting models have 96, 240, 576, and 1,344 states, respectively. The transient-state probabilities of the models are computed for the times ranging from 10 to 10,000. In all the models, the maximum of absolute diagonal elements of the generator matrices, $(\max\{E(s)\})$ is equal to 201.

Both experiments are conducted on a computer with an Intel Core i7-6700 @ 3.40 GHz. CPU, 16 GB of RAM, running Ubuntu Linux 16.04 64 bits. Both uniformization and matrix exponential have been implemented in C. Implementations do not make use of sparse matrix storages.

D. Results

The results of Experiment 1 and Experiment 2 are depicted in Fig. 1 and Fig. 2, respectively. The figures show the computation times (in milliseconds) of both methods for various sizes of models' state spaces and various values of Λt. Note that axes "Lambda * t" and "Computation Time (ms)" are in logarithmic scale.

In both figures, we can observe that computation times consistently grow for larger state-space sizes as well as for larger values of Λt. For a particular value of Λt, the computation time of both methods grows fast for larger state-space sizes, even though matrix exponential is much faster than uniformization. For a particular state-space size, the computation time of matrix exponential for different values of Λt grows slowly (less than one order of magnitude). The computation time of uniformization, for a similar situation, grows steeply; the larger Λt, the steeper the growth.

In order to more closely observe the effect of the value of Λt on the computation time, we show in Fig. 3 the computation times for increasing values of Λt of a model in Experiment 1 (containing 780 states) and a model in Experiment 2 (containing 1,344 states). We can observe that the computation time of uniformization exceeds that of matrix exponential when the value of Λt exceeds around 20,000 in the first model and around 400,000 in the second model. When Λt is less than those values, the computation time of uniformization is considerably less than that of matrix exponential. Once those values are exceeded, however, the computation time grows very fast.

E. Discussion

In this section, we have investigated how recently developed methods for computing matrix exponentials can be used in Markov chain analysis. Since transient analysis of CTMCs basically boils down to the solution of matrix exponential functions, we have compared the most promising method by Higham with uniformization, the method usually used to perform transient analysis. The comparison of both methods reveals that, at least theoretically, their computational requirements are comparable, with, as a rule of thumb, uniformization being preferable for large models and Higham's method being preferable for very stiff models.

Our practical experiments support this claim for small and moderate-size models. The advantages of matrix exponential are restricted to this range of model sizes for two reasons. First, because of the computational requirements described above. Another reason for this lies in the storage requirements of matrix exponential. Due to the heavy use of matrix multiplications in matrix exponential method, its use is restricted to models of small to moderate size. The fact that at least 4 matrices, namely the original matrix $\mathbf{Q}t$, $(\mathbf{Q}t)^2$, $(\mathbf{Q}t)^4$, and $(\mathbf{Q}t)^6$, must be stored simultaneously in memory increases the storage requirement of this method. Uniformization, on the other hand, does not suffer from this limitation.

Another advantage of uniformization lies in the *sparseness* of the model; and this allows it to scale to larger model sizes better than matrix exponential. Since CTMCs are usually structured and have a low number of non-zero elements, the use of sparse matrices or symbolic storages can considerably reduce not only the storage requirement, but also the performance. Using sparse storage, vector-matrix multiplication can be performed in $\mathcal{O}(\eta)$ multiplications, where η is the number of non-zero elements in the matrix.

Matrix exponential cannot profit from such a sparse matrix storage that much, because successive matrix multiplications result in fill-in. In current standard personal computers,

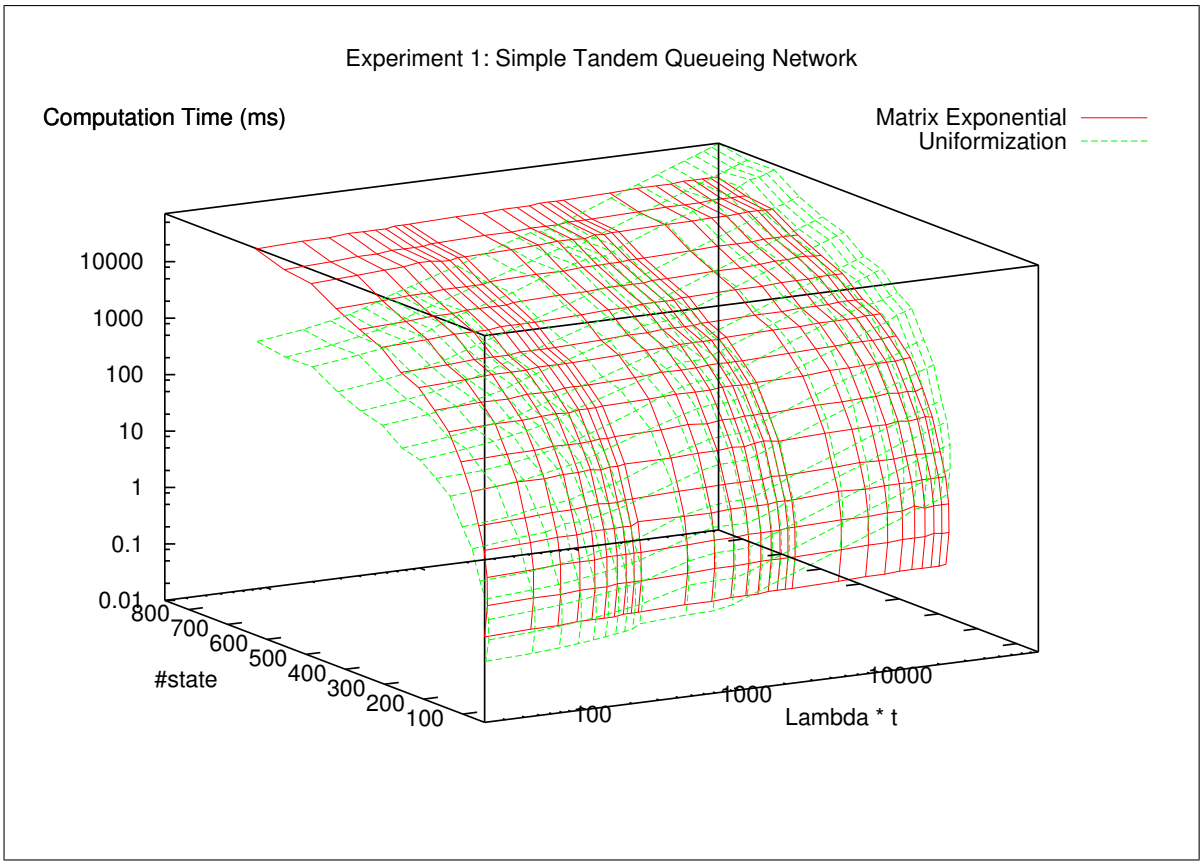

Fig. 1. A 3D chart showing the computation times (in milliseconds) of both methods for 476 cases obtained by varying the state-space size (#state) and the value of Λt for the Simple Tandem Queueing Network experiment. The state-space size ranges from 45 to 861, while Λ ranges from 22 to 86. Since the transient-state probabilities are computed for time-instant ranging from 10 to 10,000, Λt, then, ranges from 220 to 860,000. The axes for computation time and Λt are logarithmic.

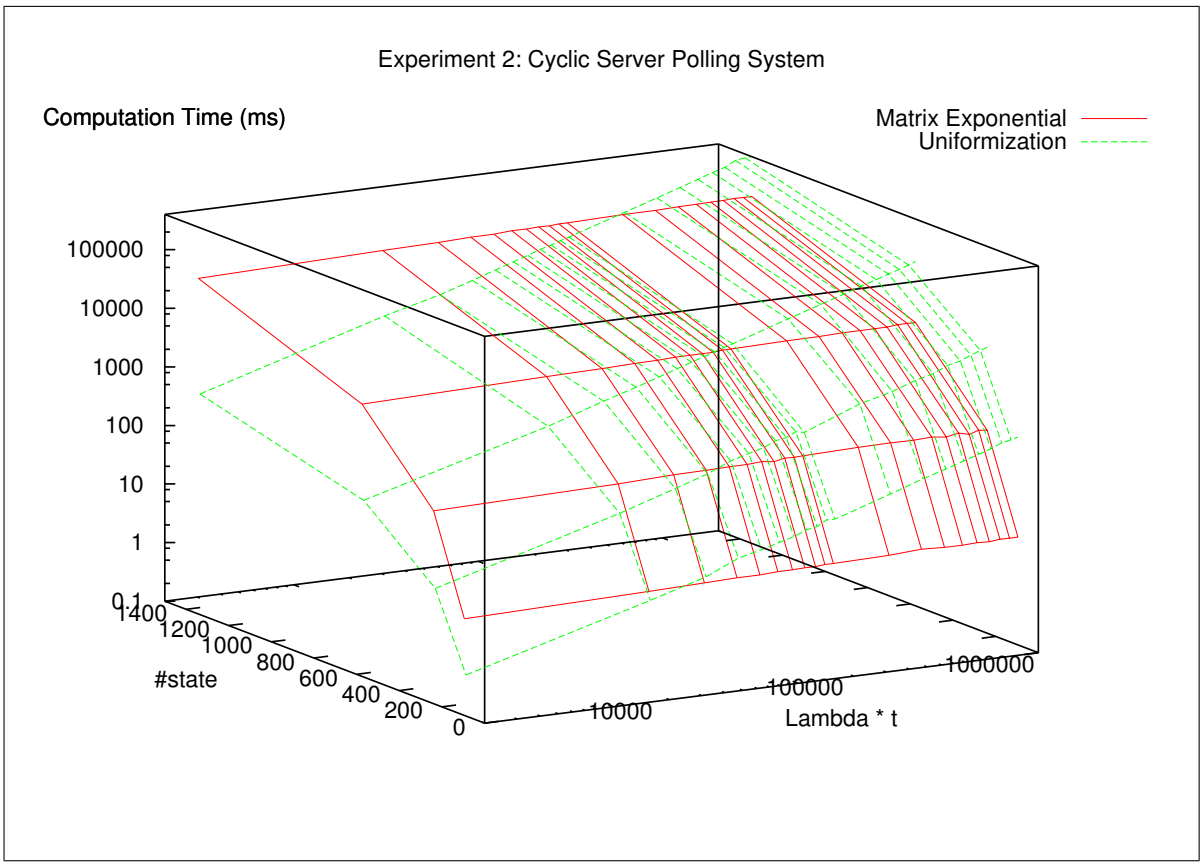

Fig. 2. A 3D chart showing the computation times (in milliseconds) of both methods for 80 cases obtained by varying the state-space size (#state) and the value of Λt for the Cyclic Server Polling System experiment. The state-space size ranges from 96 to 1,344, while Λ is fixed at 210. Since the transient-state probabilities are computed for time-instant ranging from 10 to 10,000, Λt, then, ranges from 2100 to 2,100,000. The axes for computation time and Λt are logarithmic.

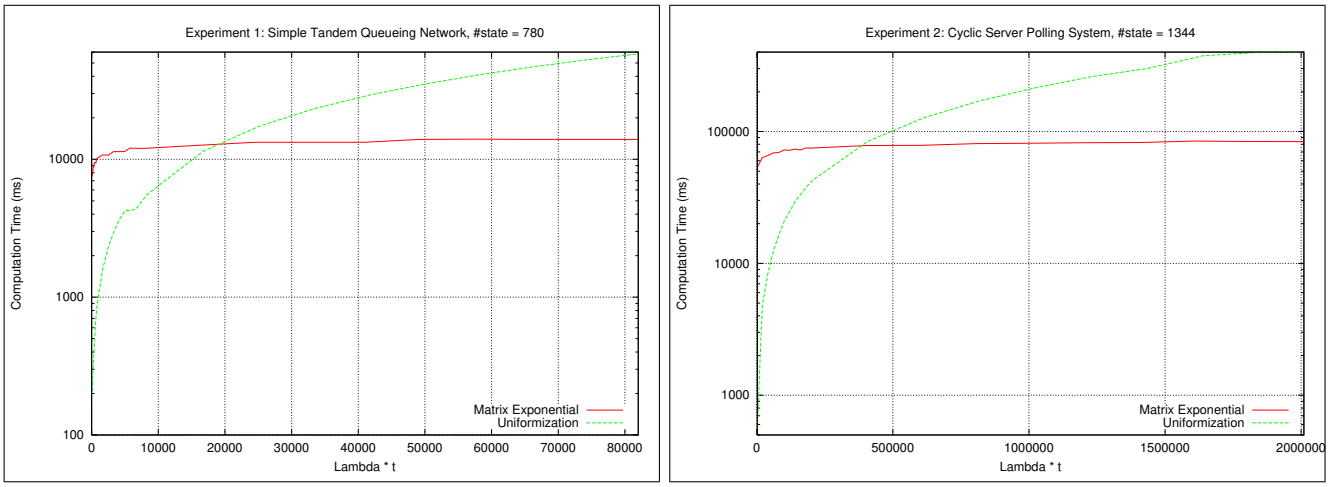

Fig. 3. Zooming into Fig. 1 and Fig. 2, this figure shows the computation times (in milliseconds) of both methods for a particular state-space size, namely 780 states for the Simple Tandem Queueing Network experiment and 1,344 states for the Cyclic Server Polling System experiment, while the value of Λt is varied. The axis for computation time is logarithmic.

matrix exponential is only practical for models with up to around 3,000 states. On the other hand, this is the range of model sizes where matrix exponential shows its strength, so there is no benefit in striving for more compact representations, which would allow it to handle models that uniformization is preferable anyhow.

With respect to numerical accuracy, we have discussed that error bounds supplied by both methods are (or can be made) comparable. In practice, for models with small index of stiffness ($\Lambda t \leq 10^2$) the performance of uniformization is still tolerable when error bound is maintained small (10^{-10} to 10^{-12}), but, as explained above, decreases with increasing index of stiffness. In Padé approximation, on the other hand, the maximum relative perturbation error is ensured not to exceed the unit round-off of IEEE double-precision arithmetic ($2^{53} \approx 1.1 \times 10^{-16}$). As a result, our experience shows that matrix exponential is safe to use for models with Λt of up to 10^5. As was to be expected, decreasing the error bound for uniformization makes the numerical results converge to those produced by matrix exponential.

As a final point, one may raise the issue that, for large values of Λt, an equilibrium might have been reached long before time-instant t. Uniformization can account for that by using a built-in steady-state detection mechanism during the evaluation of each (or some) time-step of the embedded DTMC, thus cutting computation steps. However, this equilibrium phenomenon is not to be expected for very stiff models, where even for very large time-instants (in the order of 10^3) equilibrium is still out of sight.

V. Conclusion

We have compared the performance of uniformization and matrix exponential methods in computing transient-state probabilities of CTMCs. Specifically, we have used the Padé approximation coupled with scaling and squaring method of the type proposed by Higham to compute matrix exponentials tailored to the IEEE floating-point specification.

Our experiments show that:

1) For small indices of stiffness, the computation time of uniformization is always much smaller than that of matrix exponential.
2) The index of stiffness has only a marginal effect on the computation time of matrix exponential. On uniformization, however, the index of stiffness has a significant effect.
3) The state-space size contributes much more to the computation time than the index of stiffness. Note that axis "#state" is in linear scale; and even in this scale, the growth of computation time for increasing state-space size is faster than that for increasing values of Λt, which is presented in logarithmic scale.

While in general the use of uniformization is still preferable compared to matrix exponential, very stiff models of average state-space sizes can take advantage of matrix exponential method. Such models are common in reliability engineering.

References

[1] W. Feller, *An introduction to probability theory and its applications*. John Wiley & Sons, 2008, vol. 2.
[2] J. R. Norris, *Markov Chains*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.
[3] S. M. Ross, *Stochastic processes*, ser. Wiley series in probability and statistics: Probability and statistics. Wiley, 1996.
[4] W. J. Stewart, *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.
[5] Y.-L. Tsai, D. Yanagisawa, and K. Nishinari, "Performance analysis of open queueing networks subject to breakdowns and repairs," *Engineering Letters*, vol. 24, no. 2, pp. 207–214, 2016.
[6] X. Zeng, L. Shu, and J. Jiang, "Fuzzy time series forecasting based on grey model and Markov chain," *IAENG International Journal of Applied Mathematics*, vol. 46, no. 4, pp. 464–472, 2016.
[7] R. Pulungan and H. Hermanns, "Acyclic minimality by construction—almost," in *2009 Sixth International Conference on the Quantitative Evaluation of Systems*, Sept 2009, pp. 63–72.
[8] H. R. Al-Khalidi and D. J. Schnell, "Application of a continuous-time Markov chain to a preclinical study," *Drug Information Journal*, vol. 31, no. 2, pp. 607–613, 1997.
[9] A. Jensen, "Markov chains as an aid in the study of Markov processes," *Skand. Aktuarietidskrift*, vol. 3, pp. 87–91, 1953.
[10] W. K. Grassmann, "Transient solutions in Markovian queueing systems," *Comput. & Oper. Res.*, vol. 4, no. 1, pp. 47–53, 1977.
[11] B. L. Fox and P. W. Glynn, "Computing Poisson probabilities," *Communications of the ACM*, vol. 31, no. 4, pp. 440–445, 1988.
[12] W. Grassmann, "Finding transient solutions in Markovian event systems through randomization," in *Numerical Solution of Markov Chains*, W. J. Stewart, Ed. CRC Press, 1991, ch. 18, pp. 357–371.
[13] D. Gross and D. R. Miller, "The randomization technique as a modeling tool and solution procedure for transient Markov processes," *Operations Research*, vol. 32, no. 2, pp. 343–361, 1984.

[14] J. K. Muppala and K. S. Trivedi, "Numerical transient solution of finite Markovian queueing systems," in *Queueing and Related Models*, U. N. Bhat and I. V. Basawa, Eds. Oxford University Press, 1992, ch. 18, pp. 262–284.
[15] A. Reibman and K. Trivedi, "Numerical transient analysis of Markov models," *Comput. Oper. Res.*, vol. 15, no. 1, pp. 19–36, Jan. 1988.
[16] C. Moler and C. V. Loan, "Nineteen dubious ways to compute the exponential of a matrix," *SIAM Review*, vol. 20, no. 4, pp. 801–836, 1978.
[17] ——, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review*, vol. 45, no. 1, pp. 3–49, 2003.
[18] R. B. Sidje, "Expokit: A software package for computing matrix exponentials," *ACM Trans. Math. Softw.*, vol. 24, no. 1, pp. 130–156, Mar. 1998.
[19] N. J. Higham, "The scaling and squaring method for the matrix exponential revisited," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 4, pp. 1179–1193, 2005.
[20] C. Brezinski, *Padé-Type Approximation and General Orthogonal Polynomials*, ser. International Series of Numerical Mathematics (ISNM). Birkhäuser Basel, 1980, vol. 50.
[21] "MATLAB Documentation: Matrix exponential." [Online]. Available: https://www.mathworks.com/help/matlab/ref/expm.html
[22] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *Journal of Symbolic Computation*, vol. 9, no. 3, pp. 251–280, 1990.
[23] J. K. Muppala, G. Ciardo, and K. S. Trivedi, "Stochastic reward nets for reliability prediction," *Communications in Reliability, Maintainability and Serviceability*, vol. 2, pp. 9–20, 1994.
[24] R. W. Butler, "The SURE approach to reliability analysis," *IEEE Transactions on Reliability*, vol. 41, no. 2, pp. 210–218, Jun 1992.
[25] M. Malhotra and K. S. Trivedi, "Reliability analysis of redundant arrays of inexpensive disks," *Journal of Parallel and Distributed Computing*, vol. 17, no. 1, pp. 146–151, 1993.
[26] K. S. Trivedi, *Probability & statistics with reliability, queuing and computer science applications*. John Wiley & Sons, 2008.
[27] H. Hermanns, J. Meyer-Kayser, and M. Siegle, "Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains," in *3rd Int. Workshop on the Numerical Solution of Markov Chains*, 1999, pp. 188–207.
[28] O. C. Ibe and K. S. Trivedi, "Stochastic Petri net models of polling systems," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1649–1657, Dec 1990.

Reza Pulungan received his bachelor degree in 1999 from Universitas Gadjah Mada, Indonesia, his master degree in 2004 from University of Twente, the Netherlands, and his doctoral degree in computer science in 2009 from Saarland University, Germany.

Since 2009 he has been a lecturer and a researcher at the Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada. His research interests lie in the field of stochastic processes, especially Markov processes and phase-type distributions. He is also interested in modeling and analysis of networked systems.

Holger Hermanns studied at the University of Bordeaux, France, and the University of Erlangen-Nürnberg, Germany, where he received the diploma (with honors) in computer science in 1993 and the PhD degree (with honors) from the Department of Computer Science in 1998. Since 2003 he is full professor at Saarland University, Saarbrücken, Germany, holding the chair of Dependable Systems and Software on Saarland Informatics Campus. Prior to that he held positions at the University of Twente, the Netherlands, and at INRIA Grenoble, France.

Holger Hermanns' research interests include modeling and verification of concurrent systems, resource-aware embedded systems, compositional performance and dependability evaluation, and their applications to energy informatics. He has authored or co-authored more than 200 peer-reviewed scientific papers (h-index 49, ha-index 92). He serves on the steering committees of ETAPS and TACAS, and he received the Dutch "Vernieuwingsimpuls" and the German "Preis des Fakultätentages Informatik" award. He is an ERC Advanced Grantee and an elected member of Academia Europaea.