Methodol Comput Appl Probab (2017) 19:445–485
DOI 10.1007/s11009-016-9490-y

Computing the Expected Markov Reward Rates with Stationarity Detection and Relative Error Control

Víctor Suñé[1]

Received: 13 March 2015 / Revised: 12 February 2016
Accepted: 17 February 2016 / Published online: 11 March 2016

Abstract By combining in a novel way the randomization method with the stationary detection technique, we develop two new algorithms for the computation of the expected reward rates of finite, irreducible Markov reward models, with control of the relative error. The first algorithm computes the expected transient reward rate and the second one computes the expected averaged reward rate. The algorithms are numerically stable. Further, it is argued that, from the point of view of run-time computational cost, for medium-sized and large Markov reward models, we can expect the algorithms to be better than the only variant of the randomization method that allows to control the relative error and better than the approach that consists in employing iteratively the currently existing algorithms that use the randomization method with stationarity detection but allow to control the absolute error. The performance of the new algorithms is illustrated by means of examples, showing that the algorithms can be not only faster but also more efficient than the alternatives in terms of run-time computational cost in relation to accuracy.

Keywords Markov reward model · Markov chain · Expected reward rate · Relative error · Randomization · Stationarity detection

Mathematics Subject Classification (2010) 60J28 · 60J22 · 65C40

1 Introduction

Consider a finite Markov reward model (MRM) consisting of a finite, irreducible (time homogeneous) continuous-time Markov chain (CTMC) $X = \{X(t)\,, t \geq 0\}$ with infinites-

✉ Víctor Suñé
victor.sunye@upc.edu

[1] Departament d'Enginyeria Electrònica, Universitat Politècnica de Catalunya, Colom, 1, 08222 Terrassa, Catalonia, Spain

imal generator and a reward rate vector $\mathbf{r} = (r_i)$ whose ith entry has the meaning of "rate" at which reward is earned while X is in state i. In this paper, we will be concerned with the computation of the expected transient reward rate at time t, $t > 0$, $\mathrm{ETRR}(t) = \mathrm{E}[r_{X(t)}]$, and the expected averaged reward rate in the time interval $[0, t]$, $t > 0$, $\mathrm{EARR}(t) = \mathrm{E}[(1/t)\int_0^t r_{X(\tau)}\,\mathrm{d}\tau]$. To illustrate the usefulness of ETRR(t) and EARR(t), consider a CTMC modeling a fault-tolerant system that can be up or down and assume that a reward rate 1 is assigned to the states in which the system is up and a reward rate 0 is assigned to the states in which the system is down. Then, ETRR(t) would be the availability of the system at time t, i.e., the probability that the system is up at time t, and EARR(t) would be the expected interval availability at time t, i.e., the expected fraction of the time interval $[0, t]$ in which the system is up.

We will assume that all reward rates are nonnegative and, to avoid trivialities, that at least one of them is nonnull. Since X is irreducible, this implies that, for $t > 0$, both ETRR(t) and EARR(t) are positive. The assumption that there are not negative reward rates is not a true restriction as it can be easily circumvented (Carrasco 2004).

Both ETRR(t) and EARR(t) can be computed with well-controlled error using the randomization method (also called uniformization) (Grassmann 1977; Gross and Miller 1984) and variants (van Moorsel and Sanders 1994; Sericola 1999; Carrasco 2003a, 2004; Suñé and Carrasco 2005; Sidje et al. 2007). Almost all these variants allow to compute ETRR(t) and EARR(t) with control of the absolute error, which is not always satisfactory. The reason is that if ETRR(t) and EARR(t) have to be computed with prescribed relative accuracy, then one has to use those algorithms iteratively until the accuracy requirement is fulfilled. To the best of the author's knowledge, the only variants that allow to compute ETRR(t) and EARR(t) with control of the relative error are the implementations of the randomization method developed in Suñé and Carrasco (2005). However, like most randomization-based methods, when the time t at which ETRR(t) and EARR(t) have to be computed is large, those implementations tend to have high run-time computational cost. In the case of finite, irreducible MRMs with infinitesimal generator, the run-time computational cost of the randomization method can be reduced by using the so-called stationarity detection technique. Broadly speaking, the technique consists in detecting when the underlying CTMC is close enough to its stationarity regime so that the computations can be stopped and, therefore, can result in significant reductions of the run-time computational cost when t is large. The stationarity detection technique has already been combined with the randomization method to develop algorithms for the computation of ETRR(t) and EARR(t) (Sericola 1999) which can be much faster than most randomization-based algorithms.[1] But, those algorithms allow to control the absolute error. Currently, then, to compute ETRR(t) and EARR(t) with prescribed relative accuracy, one can use the implementations of the randomization method developed in Suñé and Carrasco (2005), which, as previously commented, can be very slow when the time t is large, or else can use iteratively the algorithms developed in Sericola (1999), an approach that is not completely satisfactory either because unless a good estimate for ETRR(t) and EARR(t) is available, it can be necessary to execute the algorithms twice or more times, thus (partially) offsetting the reduction in run-time computational cost brought up by the stationarity detection technique.

In this paper, by combining in a novel way the randomization method with the stationarity detection technique proposed in Sericola (1999), we develop two new algorithms, one for the computation of ETRR(t) and another for the computation of EARR(t), with control

[1] In Sericola (1999), ETRR(t) and EARR(t) are referred to as point performability and expected interval perform ability, respectively.

of the relative error. The algorithms are numerically stable. Compared with the implementations of the randomization method developed in Suñé and Carrasco (2005) and the approach that consists in using iteratively the algorithms developed in Sericola (1999), the algorithms can be expected to have, for medium-sized and large MRMs, a lower run-time computational cost. Besides, when accuracy is taken into account, the algorithms can be substantially more efficient in the sense of being able to achieve the same accuracy with a much lower run-time computational cost. The rest of the paper is organized as follows. The algorithms are developed in Sections 2 and 3. In Section 4, we discuss the numerical stability and the run-time computational cost of the proposed algorithms. In Section 5, we illustrate the performance of the algorithms and compare them with the alternatives. Finally, in Section 6 we present some conclusions. The Appendix collects the proofs of the theoretical results on which the new algorithms are based.

2 Computation of ETRR(t)

First, we introduce some notations. We will denote by α the initial probability distribution vector of X and by $\mathbf{A} = (a_{i,j})$ its infinitesimal generator. The probability of having $j \geq 0$ arrivals in a Poisson distribution with parameter $\lambda > 0$ will be denoted $P_j(\lambda) = \mathrm{e}^{-\lambda}\lambda^j/j!$. In addition, we will denote by $\mathbf{I}$ the identity matrix, by $\lfloor x \rfloor$ the largest integer nonlarger than x, by $^{\mathrm{T}}$ the transpose operator, by $r_{\min}$ the minimal reward rate $\min_i r_i$, by $r_{\max}$ the maximal reward rate $\max_i r_i$, by ε a positive relative error tolerance, and by δ a positive quantity $\ll 1$.

Let $\Lambda \geq \max_i |a_{i,i}|$ and $\mathbf{B} = \mathbf{I} + (1/\Lambda)\mathbf{A}$ and define, for $k \geq 0$, $\mathbf{c}^{(k)} = (c_i^{(k)}) = \mathbf{B}^k(\mathbf{r}/r_{\max})$ and $v_k = \alpha^{\mathrm{T}}\mathbf{c}^{(k)}$. Using the well-known randomization result (see, e.g., Kijima 1997, Theorem 4.19), we can write

$$
\begin{aligned}
\mathrm{ETRR}(t) &= \alpha^{\mathrm{T}}\mathrm{e}^{\mathbf{A}t}\mathbf{r} \\
&= \sum_{j=0}^{\infty} \alpha^{\mathrm{T}}\mathbf{B}^j \mathrm{e}^{-\Lambda t}\frac{(\Lambda t)^j}{j!}\mathbf{r} \\
&= r_{\max}\sum_{j=0}^{\infty} \alpha^{\mathrm{T}}\mathbf{B}^j \frac{\mathbf{r}}{r_{\max}}\mathrm{e}^{-\Lambda t}\frac{(\Lambda t)^j}{j!} \\
&= r_{\max}\sum_{j=0}^{\infty} \alpha^{\mathrm{T}}\mathbf{c}^{(j)} P_j(\Lambda t) \\
&= r_{\max}\sum_{j=0}^{\infty} v_j P_j(\Lambda t)\,. \qquad (1)
\end{aligned}
$$

Let π denote the steady-state probability distribution vector of X. If $\Lambda > \max_i |a_{i,i}|$, as $k \to \infty$ each entry of the vector $\mathbf{c}^{(k)}$ tends to its stationary value $\pi^{\mathrm{T}}\mathbf{r}/r_{\max}$ (Sericola 1999). Formally, for every state i of X,

$$\lim_{k\to\infty} \mathbf{c}_i^{(k)} = \pi^{\mathrm{T}}\mathbf{r}/r_{\max}\,. \qquad (2)$$

In Sericola (1999), the above limit was turned into a practical test for stationarity detection by proving that, given the sequences $\{m_k = \min_i c_i^{(k)}\}$ and $\{M_k = \max_i c_i^{(k)}\}$, we have

$$\left| v_j - \frac{m_k + M_k}{2} \right| \leq \frac{M_k - m_k}{2}, \quad j \geq k \geq 0\,, \qquad (3)$$

and, therefore, given $k \geq 0$, the approximation for $\mathrm{ETRR}(t)$ that results from replacing in Eq. 1 all v_j, $j > k$, by $(m_k + M_k)/2$,

$$\widehat{\mathrm{ETRR}}(t,k) = r_{\max}\left(\sum_{j=0}^{k} v_j P_j(\Lambda t) + \frac{m_k + M_k}{2}\left(1 - \sum_{j=0}^{k} P_j(\Lambda t)\right)\right), \tag{4}$$

has well-controlled error

$$\left|\mathrm{ETRR}(t) - \widehat{\mathrm{ETRR}}(t,k)\right| \leq r_{\max}\frac{M_k - m_k}{2}\left(1 - \sum_{j=0}^{k} P_j(\Lambda t)\right). \tag{5}$$

The new algorithm is based on Eqs. 1 and 3. We start by noting that, since, as assumed, X is irreducible and $r_{\max} > 0$, we have

$$M_k > 0, \quad k \geq 0, \tag{6}$$

implying $m_k + M_k > 0$, $k \geq 0$. Therefore, by Eq. 4, we have $\widehat{\mathrm{ETRR}}(t,k) > 0$, $t > 0$, $k \geq 0$, and can then define the relative error incurred by approximating $\mathrm{ETRR}(t)$ by $\widehat{\mathrm{ETRR}}(t,k)$, $t > 0$, $k \geq 0$, as

$$\left|\frac{\mathrm{ETRR}(t) - \widehat{\mathrm{ETRR}}(t,k)}{\widehat{\mathrm{ETRR}}(t,k)}\right| = \frac{|\mathrm{ETRR}(t) - \widehat{\mathrm{ETRR}}(t,k)|}{\widehat{\mathrm{ETRR}}(t,k)}.$$

Trivially, to make that relative error nonlarger than ε, the index k much be such that $|\mathrm{ETRR}(t) - \widehat{\mathrm{ETRR}}(t,k)| \leq \varepsilon\widehat{\mathrm{ETRR}}(t,k)$. By Eq. 4 and Ineq. 5, to satisfy the previous inequality it is sufficient that

$$M_k - m_k \leq \left(2\frac{\sum_{j=0}^{k} v_j P_j(\Lambda t)}{1 - \sum_{j=0}^{k} P_j(\Lambda t)} + m_k + M_k\right)\varepsilon. \tag{7}$$

Since, by Eq. 2, $\lim_{k\to\infty}(M_k - m_k) = 0$, by Ineq. 6, $M_k > 0$, $k \geq 0$, and $\varepsilon > 0$, there exist infinitely many indices k satisfying Ineq. (7) and $\mathrm{ETRR}(t)$ could be computed with relative error $\leq \varepsilon$ by using Eq. 4 with k set to the minimal of those indices. However, that scheme would not be completely satisfactory because of the potential numerical cancellations involved in the computation in Eq. 4 of the term $1 - \sum_{j=0}^{k} P_j(\Lambda t)$ (Ineq. (7) can be easily rewritten to avoid the computation of that term) when the sum $\sum_{j=0}^{k} P_j(\Lambda t)$ is close to 1. To avoid those potential numerical cancellations, we will replace $1 - \sum_{j=0}^{k} P_j(\Lambda t)$ by a lower bound that does not involve significant numerical cancellations and will tighten Ineq. (7) to offset the additional error introduced by the bound.

The lower bound is

$$\left[1 - \sum_{j=0}^{k} P_j(\Lambda t)\right]^{\mathrm{lb}} = \begin{cases} 1 - \sum_{j=0}^{k} P_j(\Lambda t) & \text{if } \sum_{j=0}^{k} P_j(\Lambda t) \leq 0.9 \\ \sum_{j=k+1}^{R_1} P_j(\Lambda t) & \text{otherwise} \end{cases}, \tag{8}$$

where

$$R_1 = \min\left\{r \geq k+1 : \frac{\frac{1}{r+3-\Lambda t}\frac{r+3}{r+2}\left(r + 2 + \frac{\Lambda t}{r+3-\Lambda t}\right)P_{r+1}(\Lambda t)}{\sum_{j=k+1}^{r} P_j(\Lambda t)} \leq \delta\right\}. \tag{9}$$

Proposition 1 below shows that $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$ bounds $1 - \sum_{j=0}^{k} P_j(\Lambda t)$ from below with a relative error $\leq \delta$.

Proposition 1 *Assume $t > 0$, $k \geq 0$, and $\delta > 0$. Then, the truncation parameter R_1 given by* Eq. 9 *is finite and* $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$ *defined by* Eq. 8 *satisfies*

$$0 \leq \frac{1 - \sum_{j=0}^{k} P_j(\Lambda t) - [1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}}{1 - \sum_{j=0}^{k} P_j(\Lambda t)} \leq \delta\,. \tag{10}$$

That the computation of the lower bound $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$ will not involve significant numerical cancellations can be shown as follows. If $\sum_{j=0}^{k} P_j(\Lambda t) \leq 0.9$, then $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}} = 1 - \sum_{j=0}^{k} P_j(\Lambda t) \geq 0.1$, and, therefore, $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$ can be computed without significant numerical cancellations. If $\sum_{j=0}^{k} P_j(\Lambda t) > 0.9$, then $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}} = \sum_{j=k+1}^{R_1} P_j(\Lambda t)$ and the only subtractions are the ones involved in the computation of the term $r + 3 - \Lambda t$ in Eq. 9. Those subtractions, however, will not involve significant numerical cancellations because, using the fact that the median of a Poisson distribution with parameter Λt is nonsmaller than $\Lambda t - \log 2$ (Choi 1994), in Eq. 9 we will have $k \geq \Lambda t - \log 2$, implying $r + 3 - \Lambda t \geq k + 4 - \Lambda t \geq \Lambda t - \log 2 + 4 - \Lambda t > 3$.

If in Eq. 4 we now replace the term $1 - \sum_{j=0}^{k} P_j(\Lambda t)$ by the lower bound $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$, we obtain the new approximation

$$\widehat{\mathrm{ETRR}}(t,k) = r_{\max}\left(\sum_{j=0}^{k} v_j P_j(\Lambda t) + \frac{m_k + M_k}{2}\left[1 - \sum_{j=0}^{k} P_j(\Lambda t)\right]^{\mathrm{lb}}\right). \tag{11}$$

Let

$$\varepsilon' = \left(1 - \delta\left(1 + \frac{1}{\varepsilon}\right)\right)\varepsilon = \varepsilon(1-\delta) - \delta\,. \tag{12}$$

To offset the additional error introduced by the lower bound $[1 - \sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$, we require k in Eq. 11 to satisfy

$$M_k - m_k \leq 2\frac{\sum_{j=0}^{k} v_j P_j(\Lambda t)}{1 - \sum_{j=0}^{k} P_j(\Lambda t)}\varepsilon + (m_k + M_k)\varepsilon' \tag{13}$$

instead of Ineq. 7. Then, defining

$$K_1 = \min\{k \geq 0 : \text{ Ineq. (13) holds}\}\,, \tag{14}$$

we have the following result.

Proposition 2 *Let t, t', $0 < t' \leq t$, and assume $0 < \delta < 1$, $\varepsilon > \delta/(1-\delta)$. Then, the truncation parameter K_1 given by* Eq. 14 *is finite and $\widehat{\mathrm{ETRR}}(t', K_1)$ given by* Eq. 11 *with t replaced by t' and k replaced by K_1 satisfies*

$$\left|\frac{\mathrm{ETRR}(t') - \widehat{\mathrm{ETRR}}(t', K_1)}{\widehat{\mathrm{ETRR}}(t', K_1)}\right| \leq \varepsilon\,.$$

We are now in a position to describe the proposed algorithm for the computation of $\mathrm{ETRR}(t)$ for a set of $n, n \geq 1$, time points $0 < t_1 < \cdots < t_n$. First, we obtain the truncation parameter K_1 using Eq. 14 with t set to the largest time point t_n. To avoid the numerical cancellations potentially involved in the computation of the term $1 - \sum_{j=0}^{k} P_j(\Lambda t)$ in Ineq. (13), that inequality is used in its equivalent form

$$M_k \left(1 + \varepsilon' \sum_{j=0}^{k} P_j(\Lambda t)\right) + m_k(1+\varepsilon') \sum_{j=0}^{k} P_j(\Lambda t) \leq M_k \left(\sum_{j=0}^{k} P_j(\Lambda t) + \varepsilon'\right) + m_k(1+\varepsilon') + 2\varepsilon \sum_{j=0}^{k} v_j P_j(\Lambda t)\,. \tag{15}$$

By Proposition 2, the truncation of parameter K_1 thus obtained is such that

$$\left| \frac{\mathrm{ETRR}(t_i) - \widehat{\mathrm{ETRR}}(t_i, K_1)}{\widehat{\mathrm{ETRR}}(t_i, K_1)} \right| \leq \varepsilon$$

for all t_i, $0 < t_i \leq t_n$. Therefore, once K_1 is known, we compute $\widehat{\mathrm{ETRR}}(t_i, K_1)$, $1 \leq i \leq n$, using Eqs. 8, 9 and 11 with t replaced by t_i and k replaced by K_1. Finally, since $\mathrm{ETRR}(\infty) = \pi^{\mathrm{T}}\mathbf{r}$ and (Sericola 1999)

$$m_k \leq \pi^{\mathrm{T}}\mathbf{r}/r_{\max} \leq M_k, \quad k \geq 0\,, \tag{16}$$

we also compute the bounds $r_{\max} m_{K_1} \leq \mathrm{ETRR}(\infty) \leq r_{\max} M_{K_1}$. A detailed description of the algorithm is given on the following page.

Since the parameter δ in Algorithm 1 must be positive and the larger it is, making ε' smaller, the larger K_1 can be, in practice one will use the algorithm with δ set to a positive quantity $\ll 1$, e.g., some multiple of the machine epsilon.

To conclude this section, we note that the bounds $r_{\max} m_{K_1}$ and $r_{\max} M_{K_1}$ computed by the proposed algorithm can be used to approximate $\mathrm{ETRR}(\infty)$ by $\widehat{\mathrm{ETRR}}(\infty, K_1) = r_{\max}(m_{K_1} + M_{K_1})/2$. Since Sericola (1999) $|\mathrm{ETRR}(\infty) - r_{\max}(m_k + M_k)/2| \leq r_{\max}(M_k - m_k)/2$, $k \geq 0$, the incurred relative error will satisfy

$$\left| \frac{\mathrm{ETRR}(\infty) - \widehat{\mathrm{ETRR}}(\infty, K_1)}{\widehat{\mathrm{ETRR}}(\infty, K_1)} \right| \leq \frac{M_{K_1} - m_{K_1}}{M_{K_1} + m_{K_1}}\,. \tag{17}$$

For large enough Λt_n, we can expect that error to be close to ε. Indeed, it is easy to check that, for a fixed value of k, the function $(\sum_{j=0}^{k} v_j P_j(\Lambda t))/(1 - \sum_{j=0}^{k} P_j(\Lambda t))$ decreases to 0 as $\Lambda t \to \infty$. Therefore, by Ineq. 13 and Eq. 14, for large enough Λt_n, the parameter K_1 will be almost independent of Λt_n, satisfying $M_{K_1} - m_{K_1} \approx (m_{K_1} + M_{K_1})\varepsilon'$ and, consequently, by Ineq. 17 we will have

$$\left| \frac{\mathrm{ETRR}(\infty) - \widehat{\mathrm{ETRR}}(\infty, K_1)}{\widehat{\mathrm{ETRR}}(\infty, K_1)} \right| \leq \frac{M_{K_1} - m_{K_1}}{M_{K_1} + m_{K_1}} \approx \varepsilon' < \varepsilon\,.$$

Algorithm 1 Computation of $\mathrm{ETRR}(t)$ with control of the relative error using the randomization method with stationarity detection

input : $\mathbf{r}$, α, $\mathbf{B}$, Λ, δ, $0 < \delta < 1$, ε, $\varepsilon > \delta/(1-\delta)$, n, $n \geq 1$, $0 < t_1 < \cdots < t_n$
output : $\mathrm{ETRR}(t_i)$, $1 \leq i \leq n$; bounds for $\mathrm{ETRR}(\infty)$

$r_{\min} := \min_i r_i$;
$r_{\max} := \max_i r_i$;
$\mathbf{c}^{(0)} := \mathbf{r}/r_{\max}$;
$m_0 := r_{\min}/r_{\max}$;
$M_0 := 1$;
$v_0 := \alpha^{\mathrm{T}}\mathbf{c}^{(0)}$;
$\varepsilon' := \varepsilon(1-\delta) - \delta$;
$k := 0$;
while Ineq. (15) *with t replaced by t_n does not hold* **do**
  $k := k + 1$;
  $\mathbf{c}^{(k)} := \mathbf{B}\mathbf{c}^{(k-1)}$;
  $m_k := \min_i c_i^{(k)}$;
  $M_k := \max_i c_i^{(k)}$;
  $v_k := \alpha^{\mathrm{T}}\mathbf{c}^{(k)}$;
end
$K_1 := k$;
$i := n$;
while $i \geq 1$ **do**
  Approximate $\mathrm{ETRR}(t_i)$ by $\widehat{\mathrm{ETRR}}(t_i, K_1)$ computed using Eqs. (8), (9) and (11) with t replaced by t_i and k replaced by K_1;
  $i := i - 1$;
end
$r_{\max} m_{K_1} \leq \mathrm{ETRR}(\infty) \leq r_{\max} M_{K_1}$;

3 Computation of EARR(t)

Using $\mathrm{EARR}(t) = (1/t)\int_0^t \mathrm{ETRR}(\tau)\,\mathrm{d}\tau$, Eq. 1, and $\int_0^t P_l(\Lambda\tau)\,\mathrm{d}\tau = (1/\Lambda)\sum_{j=l+1}^{\infty} P_j(\Lambda t)$, we obtain the well-known result

$$\begin{aligned}
\mathrm{EARR}(t) &= \frac{1}{t}\int_0^t \mathrm{ETRR}(\tau)\,\mathrm{d}\tau \\
&= r_{\max}\sum_{l=0}^{\infty}\frac{1}{t}\int_0^t v_l P_l(\Lambda\tau)\,\mathrm{d}\tau \\
&= r_{\max}\sum_{l=0}^{\infty} v_l \frac{1}{\Lambda t}\sum_{j=l+1}^{\infty} P_j(\Lambda t) \\
&= r_{\max}\sum_{l=0}^{\infty} v_l \sum_{j=l}^{\infty}\frac{1}{j+1} P_j(\Lambda t)
\end{aligned}$$

$$= r_{\max} \sum_{j=0}^{\infty} \frac{1}{j+1} \sum_{l=0}^{j} v_l P_j(\Lambda t)$$
$$= r_{\max} \sum_{j=0}^{\infty} w_j P_j(\Lambda t)\,, \tag{18}$$

where

$$w_j = \frac{1}{j+1} \sum_{l=0}^{j} v_l\,. \tag{19}$$

Again, as in Sericola (1999), given some k, $k \geq 0$, we will replace all v_j, $j > k$, in Eq. 18 by $(m_k + M_k)/2$, obtaining the approximation for $\mathrm{EARR}(t)$,

$$\widehat{\mathrm{EARR}}(t,k) = r_{\max} \left(\sum_{j=0}^{k} w_j P_j(\Lambda t) + \sum_{j=k+1}^{\infty} \frac{1}{j+1} \left(\sum_{l=0}^{k} v_l + \sum_{l=k+1}^{j} \frac{m_k + M_k}{2} \right) P_j(\Lambda t) \right). \tag{20}$$

Since, by Ineq. 6, $m_k + M_k > 0$, $k \geq 0$, we have $\widehat{\mathrm{EARR}}(t,k) > 0$, $t > 0$, $k \geq 0$, and can then define the relative error incurred by approximating $\mathrm{EARR}(t)$ by $\widehat{\mathrm{EARR}}(t,k)$ as

$$\left| \frac{\mathrm{EARR}(t) - \widehat{\mathrm{EARR}}(t,k)}{\widehat{\mathrm{EARR}}(t,k)} \right| = \frac{|\mathrm{EARR}(t) - \widehat{\mathrm{EARR}}(t,k)|}{\widehat{\mathrm{EARR}}(t,k)}\,.$$

For that relative error to be nonlarger than ε, the index k must be such that

$$|\mathrm{EARR}(t) - \widehat{\mathrm{EARR}}(t,k)| \leq \varepsilon\, \widehat{\mathrm{EARR}}(t,k)\,. \tag{21}$$

For that inequality to be useful, we need computable expressions for $\widehat{\mathrm{EARR}}(t,k)$ and $|\mathrm{EARR}(t) - \widehat{\mathrm{EARR}}(t,k)|$. Let us start with $\widehat{\mathrm{EARR}}(t,k)$. By combining Eqs. 19, 20, and

$$\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t) = \sum_{j=k+1}^{\infty} \frac{j+1-(k+1)}{j+1} P_j(\Lambda t)$$
$$= \sum_{j=k+1}^{\infty} P_j(\Lambda t) - \frac{k+1}{\Lambda t} \sum_{j=k+2}^{\infty} P_j(\Lambda t)$$
$$= P_{k+1}(\Lambda t) + \frac{\Lambda t - (k+1)}{\Lambda t} \left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t) \right), \tag{22}$$

we obtain

$$\widehat{\mathrm{EARR}}(t,k) = r_{\max} \left(\sum_{j=0}^{k} w_j P_j(\Lambda t) + \sum_{j=k+1}^{\infty} \left(\frac{k+1}{j+1} w_k + \frac{j-k}{j+1} \frac{m_k + M_k}{2} \right) P_j(\Lambda t) \right)$$
$$= r_{\max} \left(\sum_{j=0}^{k} w_j P_j(\Lambda t) + \frac{k+1}{\Lambda t} w_k \sum_{j=k+2}^{\infty} P_j(\Lambda t) + \frac{m_k + M_k}{2} \sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t) \right)$$
$$= r_{\max} \left(\sum_{j=0}^{k} w_j P_j(\Lambda t) + \frac{k+1}{\Lambda t} w_k \left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t) \right) + \frac{m_k + M_k}{2} \left(P_{k+1}(\Lambda t) \right.\right.$$

$$+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)\Bigg). \tag{23}$$

With regard to $|\mathrm{EARR}(t)-\widehat{\mathrm{EARR}}(t,k)|$, using Ineq. 3 and Eqs. 18, 19, 20, and 22, we get

$$\begin{aligned}\left|\mathrm{EARR}(t)-\widehat{\mathrm{EARR}}(t,k)\right| &= \Bigg|r_{\max}\left(\sum_{j=0}^{k}w_jP_j(\Lambda t)+\sum_{j=k+1}^{\infty}\frac{1}{j+1}\left(\sum_{l=0}^{k}v_l+\sum_{l=k+1}^{j}v_l\right)P_j(\Lambda t)\right)\\ &\quad -r_{\max}\left(\sum_{j=0}^{k}w_jP_j(\Lambda t)+\sum_{j=k+1}^{\infty}\frac{1}{j+1}\left(\sum_{l=0}^{k}v_l+\sum_{l=k+1}^{j}\frac{m_k+M_k}{2}\right)P_j(\Lambda t)\right)\Bigg|\\ &= r_{\max}\left|\sum_{j=k+1}^{\infty}\frac{1}{j+1}\sum_{l=k+1}^{j}\left(v_l-\frac{m_k+M_k}{2}\right)P_j(\Lambda t)\right|\\ &\le r_{\max}\sum_{j=k+1}^{\infty}\frac{1}{j+1}\sum_{l=k+1}^{j}\left|v_l-\frac{m_k+M_k}{2}\right|P_j(\Lambda t)\\ &\le r_{\max}\sum_{j=k+1}^{\infty}\frac{1}{j+1}\sum_{l=k+1}^{j}\frac{M_k-m_k}{2}P_j(\Lambda t)\\ &= r_{\max}\frac{M_k-m_k}{2}\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)\\ &= r_{\max}\frac{M_k-m_k}{2}\left(P_{k+1}(\Lambda t)+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)\right). \end{aligned}\tag{24}$$

Then, using Eqs. 23 and 24, it is easily seen that Ineq. (21) holds for any index k satisfying

$$M_k-m_k\le\left(2\frac{\sum_{j=0}^{k}w_jP_j(\Lambda t)+\frac{k+1}{\Lambda t}w_k\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)}{P_{k+1}(\Lambda t)+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)}+m_k+M_k\right)\varepsilon\,. \tag{25}$$

However, in order to obtain a simpler algorithm, we will consider the inequality

$$M_k-m_k\le\left(2\frac{\sum_{j=0}^{k}w_jP_j(\Lambda t)}{P_{k+1}(\Lambda t)+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)}+m_k+M_k\right)\varepsilon\,, \tag{26}$$

which, since $1-\sum_{j=0}^{k+1}P_j(\Lambda t)>0$, is more restrictive than Ineq. 25.

Since, by Eq. 2, $\lim_{k\to\infty}(M_k-m_k)=0$, by Ineq. 6, $M_k>0$, and $\varepsilon>0$, there are infinitely many indices k satisfying Ineq. 26 and $\mathrm{EARR}(t)$ could be computed with relative error $\le\varepsilon$ by using Eq. 23 with k set to the minimal of those indices. This, however, could be problematic because of the numerical cancellations potentially involved in the computation of the terms $1-\sum_{j=0}^{k+1}P_j(\Lambda t)$ and $P_{k+1}(\Lambda t)+((\Lambda t-(k+1))/(\Lambda t))(1-\sum_{j=0}^{k+1}P_j(\Lambda t))$ (Ineq. (26) can be easily rewritten to avoid the computation of the latter term). To avoid those potential numerical cancellations, we will replace those terms by appropriate lower bounds not involving significant numerical cancellations and will modify Ineq. (26) appropriately.

The term $1-\sum_{j=0}^{k+1} P_j(\Lambda t)$ will be replaced by

$$\left[1-\sum_{j=0}^{k+1} P_j(\Lambda t)\right]^{\mathrm{lb}} = \begin{cases} 1-\sum_{j=0}^{k+1} P_j(\Lambda t) & \text{if } \sum_{j=0}^{k+1} P_j(\Lambda t) \le 0.9 \\ \sum_{j=k+2}^{R_2} P_j(\Lambda t) & \text{otherwise} \end{cases}, \tag{27}$$

where

$$R_2 = \min\left\{ r \ge k+2 : \frac{\frac{1}{r+3-\Lambda t}\frac{r+3}{r+2}\left(r+2+\frac{\Lambda t}{r+3-\Lambda t}\right) P_{r+1}(\Lambda t)}{\sum_{j=k+2}^{r} P_j(\Lambda t)} \le \delta \right\}, \tag{28}$$

and, recalling Eq. 22, the term $P_{k+1}(\Lambda t) + ((\Lambda t-(k+1))/(\Lambda t))(1-\sum_{j=0}^{k+1} P_j(\Lambda t)) = \sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)$ will be replaced by

$$\left[\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t)\right]^{\mathrm{lb}} = \begin{cases} P_{k+1}(\Lambda t) + \dfrac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1} P_j(\Lambda t)\right) & \text{if } k+1 \le \Lambda t \text{ or else } 1- 0.9\frac{\Lambda t}{k+1-\Lambda t} P_{k+1}(\Lambda t) \le \sum_{j=0}^{k+1} P_j(\Lambda t) \le 0.9 \\ \sum_{j=k+1}^{R_3} \frac{j-k}{j+1} P_j(\Lambda t) & \text{otherwise} \end{cases}, \tag{29}$$

where

$$R_3 = \min\left\{ r \ge k+1 : \frac{\frac{1}{r+3-\Lambda t}\frac{r+3}{r+2}\left(r+1-k+\frac{\Lambda t}{r+3-\Lambda t}\right) P_{r+1}(\Lambda t)}{\sum_{j=k+1}^{r} \frac{j-k}{j+1} P_j(\Lambda t)} \le \delta \right\}. \tag{30}$$

The factor $\Lambda t-(k+1)$ in Eq. 29 will be computed accurately by casting it as the dot product $(\Lambda, k+1)(t, -1)$ and computing that dot product using Algorithm 5.3 in Ogita et al. (2005). By Proposition 3 below, $[1-\sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}$ bounds $1-\sum_{j=0}^{k+1} P_j(\Lambda t)$ from below with a relative error $\le \delta$ and $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$ bounds $P_{k+1}(\Lambda t)+((\Lambda t-(k+1))/(\Lambda t))(1-\sum_{j=0}^{k+1} P_j(\Lambda t))$ from below with a relative error $\le \delta$.

Proposition 3 *Let $t>0$, $k \ge 0$, and $\delta > 0$. Then, the truncation parameters R_2 and R_3 given by, respectively,* Eqs. 28 *and* 30 *are finite,* $[1-\sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}$ *defined by* Eq. 27 *satisfies*

$$0 \le \frac{1-\sum_{j=0}^{k+1} P_j(\Lambda t) - [1-\sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}}{1-\sum_{j=0}^{k+1} P_j(\Lambda t)} \le \delta, \tag{31}$$

and $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$ *defined by* Eq. 29 *satisfies*

$$0 \le \frac{P_{k+1}(\Lambda t) + \dfrac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1} P_j(\Lambda t)\right) - \left[\sum_{j=k+1}^{\infty} \dfrac{j-k}{j+1} P_j(\Lambda t)\right]^{\mathrm{lb}}}{P_{k+1}(\Lambda t) + \dfrac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1} P_j(\Lambda t)\right)} \le \delta. \tag{32}$$

That the computation of the lower bound $[1-\sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}$ will not involve significant numerical cancellations can be shown similarly as it has been done for the computation of the lower bound $[1-\sum_{j=0}^{k} P_j(\Lambda t)]^{\mathrm{lb}}$ defined by Eq. 8. To show that the computation of the lower bound $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$ will not involve significant numerical cancellations, we will consider three cases separately: a) $k+1 \le \Lambda t$; b) $k+1 > \Lambda t$, $1-(0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t) \le \sum_{j=0}^{k+1} P_j(\Lambda t) \le 0.9$; and c) otherwise, i.e., $k+$

$1 > \Lambda t$, and $1 - (0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t) > \sum_{j=0}^{k+1} P_j(\Lambda t)$ or $\sum_{j=0}^{k+1} P_j(\Lambda t) > 0.9$. In case a), which can only happen if $\Lambda t \geq 1$, the only possible source of numerical cancellations is the computation of the term $1 - \sum_{j=0}^{k+1} P_j(\Lambda t)$ in Eq. 29. (We recall that the factor $\Lambda t - (k+1)$ will be computed accurately.) Using the fact that the median of a Poisson distribution with parameter Λt is nonsmaller than $\Lambda t - \log 2$ (Choi 1994) and noting that, trivially, $\lfloor \Lambda t \rfloor - 1 < \lfloor \Lambda t \rfloor - \log 2 \leq \Lambda t - \log 2$, we will have $\sum_{j=0}^{\lfloor \Lambda t \rfloor - 1} P_j(\Lambda t) < 0.5$ and, therefore, for $k+1 \leq \Lambda t$, or, equivalently, $k+1 \leq \lfloor \Lambda t \rfloor$,

$$1 - \sum_{j=0}^{k+1} P_j(\Lambda t) \geq 1 - \sum_{j=0}^{\lfloor \Lambda t \rfloor} P_j(\Lambda t) = 1 - \sum_{j=0}^{\lfloor \Lambda t \rfloor - 1} P_j(\Lambda t) - P_{\lfloor \Lambda t \rfloor}(\Lambda t) > 0.5 - P_{\lfloor \Lambda t \rfloor}(\Lambda t)\,. \quad (33)$$

Now, since $\Lambda t \geq 1$, either $\lfloor \Lambda t \rfloor = 1$ or $\lfloor \Lambda t \rfloor \geq 2$. If $\lfloor \Lambda t \rfloor = 1$, $P_{\lfloor \Lambda t \rfloor}(\Lambda t) = P_1(\Lambda t) = \Lambda t\, \mathrm{e}^{-\Lambda t}$, which reaches its maximum at $\Lambda t = 1$ and is therefore nonlarger than $P_1(1) = \mathrm{e}^{-1} < 0.4$, implying, by Ineq. 33, $1 - \sum_{j=0}^{k+1} P_j(\Lambda t) > 0.5 - 0.4 = 0.1$. If $\lfloor \Lambda t \rfloor \geq 2$, we have (Glynn 1987) $P_{\lfloor \Lambda t \rfloor}(\Lambda t) \leq 1/\sqrt{2\pi \lfloor \Lambda t \rfloor}$, which, for $\lfloor \Lambda t \rfloor \geq 2$, is < 0.3, implying, by Ineq. 33, $1 - \sum_{j=0}^{k+1} P_j(\Lambda t) > 0.5 - 0.3 = 0.2$. This shows that, in case a), the term $1 - \sum_{j=0}^{k+1} P_j(\Lambda t)$ and, therefore, the lower bound $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$, can be computed without significant numerical cancellations. In case b), in which $k+1 > \Lambda t$, $1 - (0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t) \leq \sum_{j=0}^{k+1} P_j(\Lambda t)$, and $\sum_{j=0}^{k+1} P_j(\Lambda t) \leq 0.9$, we will have $1 - \sum_{j=0}^{k+1} P_j(\Lambda t) \geq 0.1$. This implies that the only possible source of significant numerical cancellations lies in the subtraction of the quantity $((k+1-\Lambda t)/(\Lambda t))(1 - \sum_{j=0}^{k+1} P_j(\Lambda t))$ from the probability $P_{k+1}(\Lambda t)$ in Eq. 29. But, it can be seen that $1 - (0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t) \leq \sum_{j=0}^{k+1} P_j(\Lambda t)$ implies $P_{k+1}(\Lambda t) + ((\Lambda t - (k+1))/(\Lambda t))(1 - \sum_{j=0}^{k+1} P_j(\Lambda t) \geq 0.1 P_{k+1}(\Lambda t)$. Consequently, in case b), the lower bound $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$ can also be computed without significant numerical cancellations. It remains to discuss case c). In that case, $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}} = \sum_{j=k+1}^{R_3}((j-k)/(j+1))P_j(\Lambda t)$ and the only possible source of significant numerical cancellations lies in the subtractions involved in the computation of the term $r + 3 - \Lambda t$ in Eq. 30. (The term $r + 1 - k$ involves only integers and can be computed exactly.) But, those subtractions will not involve significant numerical cancellations because we will have $r + 3 - \Lambda t \geq k + 1 + 3 - \Lambda t > 3$. This concludes the justification that the computation of the lower bound $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$ will not involve significant numerical cancellations.

Having defined the bounds, in Eq. 23 we replace $1 - \sum_{j=0}^{k+1} P_j(\Lambda t)$ by $[1 - \sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}$ and $P_{k+1}(\Lambda t) + ((\Lambda t - (k+1))/(\Lambda t))(1 - \sum_{j=0}^{k+1} P_j(\Lambda t))$ by $[\sum_{j=k+1}^{\infty}((j-k)/(j+1))P_j(\Lambda t)]^{\mathrm{lb}}$, obtaining the new approximation

$$\begin{aligned}\widehat{\mathrm{EARR}}(t,k) \;=\; & r_{\max}\sum_{j=0}^{k} w_j P_j(\Lambda t) + r_{\max}\frac{k+1}{\Lambda t} w_k \left[1 - \sum_{j=0}^{k+1} P_j(\Lambda t)\right]^{\mathrm{lb}} \\ & + r_{\max}\frac{m_k + M_k}{2}\left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)\right]^{\mathrm{lb}}, \end{aligned} \quad (34)$$

and, to offset the additional errors introduced by the bounds, require k to satisfy

$$M_k - m_k \leq 2\frac{\sum_{j=0}^{k} w_j P_j(\Lambda t)}{P_{k+1}(\Lambda t) + \frac{\Lambda t - (k+1)}{\Lambda t}\left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t)\right)}\varepsilon + (m_k + M_k)\varepsilon' \quad (35)$$

instead of Ineq. 26, where ε' is given by Eq. 12. Then, defining

$$K_2 = \min\{k \geq 0 : \text{ Inequality (35) holds}\}\ , \tag{36}$$

we have the following result.

Proposition 4 *Let* $0 < t' \leq t$ *and assume* $0 < \delta < 1$, $\varepsilon > \delta/(1-\delta)$. *Then, the truncation parameter* K_2 *given by* Eq. 36 *is finite and* $\widehat{\text{EARR}}(t', K_2)$ *given by* Eq. 34 *with* t *replaced by* t' *and* k *replaced by* K_2 *satisfies*

$$\left|\frac{\text{EARR}(t') - \widehat{\text{EARR}}(t', K_2)}{\widehat{\text{EARR}}(t', K_2)}\right| \leq \varepsilon\,.$$

We are now in a position to describe the proposed algorithm for the computation of EARR(t) for a set of n, $n \geq 1$, time points $0 < t_1 < \cdots < t_n$. First, we obtain the truncation parameter K_2 using Eq. 36 with t set to the largest time point t_n, where the weights w_k, $k \geq 0$, are computed using $w_0 = v_0$ and $w_k = (1/(k+1))(kw_{k-1} + v_k)$, $k \geq 1$. To avoid the numerical cancellations potentially involved in the computation of the term $P_{k+1}(\Lambda t) + ((\Lambda t - (k+1))/(\Lambda t))(1 - \sum_{j=0}^{k+1} P_j(\Lambda t))$ in Ineq. 35, that inequality is used in its equivalent form

$$\begin{aligned}
&M_k\left(P_{k+1}(\Lambda t) + 1 + \frac{k+1}{\Lambda t}\left(\sum_{j=0}^{k+1} P_j(\Lambda t) + \varepsilon'\right) + \varepsilon' \sum_{j=0}^{k+1} P_j(\Lambda t)\right)\\
&+m_k\left(\sum_{j=0}^{k+1} P_j(\Lambda t) + \frac{k+1}{\Lambda t}\right)(1+\varepsilon')\\
\leq\ & M_k\left(\sum_{j=0}^{k+1} P_j(\Lambda t) + \frac{k+1}{\Lambda t} + \varepsilon'\left(P_{k+1}(\Lambda t) + 1 + \frac{k+1}{\Lambda t}\sum_{j=0}^{k+1} P_j(\Lambda t)\right)\right)\\
&+m_k\left(1 + P_{k+1}(\Lambda t) + \frac{k+1}{\Lambda t}\sum_{j=0}^{k+1} P_j(\Lambda t)\right)(1+\varepsilon') + 2\sum_{j=0}^{k} w_j P_j(\Lambda t)\,\varepsilon\,.
\end{aligned} \tag{37}$$

By Proposition 4, the truncation parameter K_2 thus obtained is such that

$$\left|\frac{\text{EARR}(t_i) - \widehat{\text{EARR}}(t_i, K_2)}{\widehat{\text{EARR}}(t_i, K_2)}\right| \leq \varepsilon$$

for all t_i, $0 < t_i \leq t_n$. Therefore, once K_2 is known, we compute $\widehat{\text{EARR}}(t_i, K_2)$, $1 \leq i \leq n$, using Eqs. 27, 28, 29, 30, and 34 with t replaced by t_i and k replaced by K_2. Finally, since EARR$(\infty) = \boldsymbol{\pi}^{\mathrm{T}}\mathbf{r}$ and Eq. 16, we also compute the bounds $r_{\max} m_{K_2} \leq \text{EARR}(\infty) \leq r_{\max} M_{K_2}$. A detailed description of the algorithm is given on the next page.

In practice, Algorithm 2 will be used with δ set to a positive quantity $\ll 1$, e.g., some multiple of the machine epsilon.

To conclude this section, we note that the bounds $r_{\max} m_{K_2}$ and $r_{\max} M_{K_2}$ computed by Algorithm 2 can be used to approximate EARR(∞) by $\widehat{\text{EARR}}(\infty, K_2) = r_{\max}(m_{K_2} + M_{K_2})/2$. Since EARR$(\infty)$ = ETRR(∞), $\widehat{\text{EARR}}(\infty, K_2) = \widehat{\text{ETRR}}(\infty, K_2)$, and Ineq. 17, the relative error incurred by that approximation will satisfy

$$\left|\frac{\text{EARR}(\infty) - \widehat{\text{EARR}}(\infty, K_2)}{\widehat{\text{EARR}}(\infty, K_2)}\right| \leq \frac{M_{K_2} - m_{K_2}}{M_{K_2} + m_{K_2}}\,. \tag{38}$$

Algorithm 2 Computation of EARR(t) with control of the relative error using the randomization method with stationarity detection

input : $\mathbf{r}$, α, $\mathbf{B}$, Λ, δ, $0 < \delta < 1$, ε, $\varepsilon > \delta/(1-\delta)$, n, $n \geq 1$, $0 < t_1 < \cdots < t_n$
output : EARR(t_i), $1 \leq i \leq n$; bounds for EARR(∞)

$r_{\min} := \min_i r_i$;
$r_{\max} := \max_i r_i$;
$\mathbf{c}^{(0)} := \mathbf{r}/r_{\max}$;
$m_0 := r_{\min}/r_{\max}$;
$M_0 := 1$;
$v_0 := \alpha^{\mathrm{T}}\mathbf{c}^{(0)}$;
$w_0 := v_0$;
$\varepsilon' := \varepsilon(1-\delta) - \delta$;
$k := 0$;
while Ineq. (37) *with t replaced by t_n does not hold* **do**
  $k := k + 1$;
  $\mathbf{c}^{(k)} := \mathbf{B}\mathbf{c}^{(k-1)}$;
  $m_k := \min_i c_i^{(k)}$;
  $M_k := \max_i c_i^{(k)}$;
  $v_k := \alpha^{\mathrm{T}}\mathbf{c}^{(k)}$;
  $w_k := (k/(k+1))w_{k-1} + (1/(k+1))v_k$;
end
$K_2 := k$;
$i := n$;
while $i \geq 1$ **do**
  Approximate EARR(t_i) by $\widehat{\mathrm{EARR}}(t_i, K_2)$ computed using Eqs. (27), (28), (29), (30), and (34) with t replaced by t_i and k replaced by K_2, where, if $\sum_{j=0}^{K_2+1} P_j(\Lambda t_i) > 0.9$ and $K_2 + 1 > \Lambda t_i$, the truncation parameters R_2 and R_3 are computed simultaneously to save Poisson probabilities;
  $i := i - 1$;
end
$r_{\max} m_k \leq \mathrm{EARR}(\infty) \leq r_{\max} M_k$;

For large enough Λt_n, we can expect that error to be close to ε. Indeed, by Eq. 22 we have

$$\frac{\sum_{j=0}^{k} w_j P_j(\Lambda t)}{P_{k+1}(\Lambda t) + \frac{\Lambda t-(k+1)}{\Lambda t}\left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t)\right)} = \frac{\sum_{j=0}^{k} w_j P_j(\Lambda t)}{\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t)},$$

a function that decreases to 0 as $\Lambda t \to \infty$. Therefore, by Ineq. 35 and Eq. 36, for large enough Λt_n, the parameter K_2 will be almost independent of Λt_n, satisfying $M_{K_2} - m_{K_2} \approx (m_{K_2} + M_{K_2})\varepsilon'$ and, then, by Ineq. 38, we will have

$$\left| \frac{\mathrm{EARR}(\infty) - \widehat{\mathrm{EARR}}(\infty, K_2)}{\widehat{\mathrm{EARR}}(\infty, K_2)} \right| \leq \frac{M_{K_2} - m_{K_2}}{M_{K_2} + m_{K_2}} \approx \varepsilon' < \varepsilon \, .$$

4 Numerical Stability and Run-Time Computational Cost

In this section, we will analyze the numerical stability and run-time computational cost of the proposed algorithms. We will also argue that, for medium-sized and large MRMs, we can expect the run-time computational cost of the proposed algorithms to be lower than that of the implementations of the randomization method developed in Suñé and Carrasco (2005) and lower than the run-time computational cost of the approach that consists in using iteratively the algorithms developed in Sericola (1999).

Not involving more subtractions than those required for the computation of the lower bounds defined by Eqs. 8, 27, and 29, which, as argued in Sections 2 and 3, should not result in significant numerical cancellations, and assuming that the involved Poisson probabilities are computed using a method with good numerical properties such as the one described in (Knüsel 1986, pp. 1028–1029) (see also Abramowitz and Stegun 1964), or, for sums of the form $\sum_{j=0}^{n} P_j(\Lambda t)$, the method described in Bowerman et al. (1990), the proposed algorithms are numerically stable.

For medium-sized and large MRMs, we can expect the run-time computational cost of the proposed algorithms to be dominated by the matrix-vector multiplies (MVMs) with matrix $\mathbf{B}$. For Algorithm 1, the number of such MVMs will be equal to the value of the truncation parameter K_1 defined by Eq. 14 with t replaced by t_n and, for Algorithm 2, it will be equal to the value of the truncation parameter K_2 defined by Eq. 36 with t replaced by t_n. However, in general it seems difficult to anticipate the values of K_1 or K_2.

The implementation of the randomization method developed in Suñé and Carrasco (2005) for the computation of ETRR(t), which will be referred to as Algorithm SC1, allows to compute ETRR(t) for a set of time points $t_1 < \cdots < t_n$ and will involve a number of MVMs with matrix $\mathbf{B}$ equal to

$$K_1' = \min\left\{k \geq 0 :\ \frac{1 - \sum_{j=0}^{k} P_j(\Lambda t_n)}{\sum_{j=0}^{k} v_j P_j(\Lambda t_n)} \leq \frac{\varepsilon}{4}\right\} . \tag{39}$$

The implementation of the randomization method developed in Suñé and Carrasco (2005) for the computation of EARR(t), which will be referred to as Algorithm SC2, allows to compute EARR(t) for a set of time points $t_1 < \cdots < t_n$ and will involve a number of MVMs with matrix $\mathbf{B}$ equal to

$$K_2' = \min\left\{k \geq 0 :\ \frac{1 - \sum_{j=0}^{k} P_j(\Lambda t_n)}{\sum_{j=0}^{k} w_j P_j(\Lambda t_n)} \leq \frac{\varepsilon}{4}\right\} . \tag{40}$$

For medium-sized and large MRMs, we can expect those MVMs to dominate the run-time computational cost of Algorithms SC1 and SC2. But, by Ineq. 13 and Eq. 14, using that, as assumed, $\varepsilon > \delta/(1-\delta)$, the truncation parameter K_1 of Algorithm 1 will satisfy

$$K_1 = \min\left\{k \geq 0 :\ (M_k - m_k)\frac{1 - \sum_{j=0}^{k} P_j(\Lambda t_n)}{\sum_{j=0}^{k} v_j P_j(\Lambda t_n)} \leq \varepsilon_1\right\}$$

with

$$\varepsilon_1 = \left(2 + (m_k + M_k)\frac{1 - \sum_{j=0}^{k} P_j(\Lambda t_n)}{\sum_{j=0}^{k} v_j P_j(\Lambda t_n)}(1 - \delta(1 + 1/\varepsilon))\right)\varepsilon > 2\varepsilon ,$$

and, by Eq. 22, Ineq. 35, and Eq. 36, the truncation parameter K_2 of Algorithm 2 will satisfy

$$K_2 = \min\left\{k \geq 0 :\ (M_k - m_k)\frac{1 - \sum_{j=0}^{k} P_j(\Lambda t_n)}{\sum_{j=0}^{k} w_j P_j(\Lambda t_n)} \leq \varepsilon_2\right\}$$

with

$$\varepsilon_2 = \left(2 + (m_k + M_k)\frac{\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t_n)}{\sum_{j=0}^{k} w_j P_j(\Lambda t_n)}(1-\delta(1+1/\varepsilon)) + \frac{M_k - m_k}{\varepsilon}\frac{k+1}{\Lambda t}\right.$$
$$\left.\times\frac{1-\sum_{j=0}^{k+1}P_j(\Lambda t_n)}{\sum_{j=0}^{k} w_j P_j(\Lambda t_n)}\right)\varepsilon > 2\varepsilon\,.$$

Also, since $0 \le 1 - r_{\min}/r_{\max} = M_0 - m_0 \le 1$ and the sequences $\{M_k\}$ and $\{m_k\}$ are, respectively, nonincreasing and nondecreasing (Sericola 1999), it follows that $0 \le M_k - m_k \le 1$, $k \ge 0$. Therefore, we will always have $K_1 \le K_1'$ and $K_2 \le K_2'$. Further, for large enough Λt_n, we will have $\sum_{j=0}^{K_1'} v_j P_j(\Lambda t_n) \approx \mathrm{ETRR}(\infty)$ and $\sum_{j=0}^{K_2'} w_j P_j(\Lambda t_n) \approx \mathrm{EARR}(\infty)$. Thus, for large enough Λt_n, both K_1' defined by Eq. 39 and K_2' defined by Eq. 40 will depend, essentially, on the course of $1 - \sum_{j=0}^{k} P_j(\Lambda t_n)$. But, for $\Lambda t_n \to \infty$, a Poisson distribution with parameter Λt_n has an asymptotic normal distribution with mean and variance Λt_n. Then, for large enough Λt_n and $\varepsilon \ll 1$, the parameters K_1' and K_2' will be of the order of Λt_n. On the other hand, as already argued at the end of Sections 2 and 3, for large enough Λt_n, the parameters K_1 and K_2 will be almost independent of Λt_n. Therefore, for large enough Λt_n, the differences $K_1' - K_1$ and $K_2' - K_2$ will increase with Λt_n. In summary, for medium-sized and large MRMs, we can expect the run-time computational cost of Algorithm 1 to be lower than that of Algorithm SC1, can expect the run-time computational cost of Algorithm 2 to be lower than that of Algorithm SC2, and the larger Λt_n, the larger we can expect to be the difference in run-time computational cost between Algorithms 1 and SC1 and between Algorithms 2 and SC2.

Let us now describe reasonable schemes to compute $\mathrm{ETRR}(t_i)$ and $\mathrm{EARR}(t_i)$, $1 \le i \le n$, with control of the relative error based on using iteratively the two algorithms developed in Sericola (1999). The first such algorithm computes $\mathrm{ETRR}(t_i)$, $1 \le i \le n$, with control of the absolute error and the second one computes $\mathrm{EARR}(t_i)$, $1 \le i \le n$, with control of the absolute error. Therefore, $\mathrm{ETRR}(t_i)$, $1 \le i \le n$, can be computed by invoking the first algorithm with an absolute error tolerance $\nu^{(1)} = \varepsilon(r_{\min} + r_{\max})/2$, say, and, next, if $\nu^{(1)}/\min_{1\le i\le n}\widehat{\mathrm{ETRR}}^{(1)}(t_i) > \varepsilon$, where $\widehat{\mathrm{ETRR}}^{(j)}(t_i)$ denotes the approximation for $\mathrm{ETRR}(t_i)$ computed by the algorithm in the course of invocation j with the convention $\widehat{\mathrm{ETRR}}^{(0)}(t_i) = (r_{\min} + r_{\max})/2$, continue invoking iteratively the algorithm with an absolute error tolerance

$$\nu^{(j)} = \nu^{(j-1)}\min\left\{\frac{\min_{1\le i\le n}\widehat{\mathrm{ETRR}}^{(j-1)}(t_i)}{\min_{1\le i\le n}\widehat{\mathrm{ETRR}}^{(j-2)}(t_i)}, 0.95\right\}$$

for invocation j, $j > 1$, until $\nu^{(j)}/\min_{1\le i\le n}\widehat{\mathrm{ETRR}}^{(j)}(t_i) \le \varepsilon$. (The 0.95 is a security factor to help ensure convergence.) Similarly, in the iterative scheme for the computation of $\mathrm{EARR}(t_i)$, $1 \le i \le n$, we can start by invoking the second algorithm developed in Sericola (1999) with an absolute error tolerance $\nu^{(1)} = \varepsilon(r_{\min} + r_{\max})/2$ and, next, if $\nu^{(1)}/\min_{1\le i\le n}\widehat{\mathrm{EARR}}^{(1)}(t_i) > \varepsilon$, where $\widehat{\mathrm{EARR}}^{(j)}(t_i)$ denotes the approximation for $\mathrm{EARR}(t_i)$ computed by the algorithm in the course of invocation j with the convention $\widehat{\mathrm{EARR}}^{(0)}(t_i) = (r_{\min} + r_{\max})/2$, continue invoking iteratively the algorithm with an absolute error tolerance

$$\nu^{(j)} = \nu^{(j-1)}\min\left\{\frac{\min_{1\le i\le n}\widehat{\mathrm{EARR}}^{(j-1)}(t_i)}{\min_{1\le i\le n}\widehat{\mathrm{EARR}}^{(j-2)}(t_i)}, 0.95\right\}$$

for invocation j, $j > 1$, until $\nu^{(j)}/\min_{1\le i\le n}\widehat{\mathrm{EARR}}^{(j)}(t_i) \le \varepsilon$. For the sake of conciseness, the iterative schemes for the computation of $\mathrm{ETRR}(t)$ and $\mathrm{EARR}(t)$ just described will be referred to as Algorithm SE1 and Algorithm SE2, respectively. Let S_1 denote the number of times that the first algorithm developed in Sericola (1999) is invoked in Algorithm SE1. Invocation j, $1 \le j \le S_1$, of that algorithm will involve a number of MVMs with matrix $\mathbf{B}$ equal to $\min\{N^{(j)}, K^{(j)}\}$, where

$$N^{(j)} = \min\left\{n \ge 0 : r_{\max}\left(1 - \sum_{j=0}^{n} P_j(\Lambda t_n)\right) \le \nu^{(j)}\right\}, \qquad (41)$$

$$K^{(j)} = \min\left\{k \ge 0 : r_{\max}(M_k - m_k) \le \nu^{(j)}/2\right\}. \qquad (42)$$

Therefore, Algorithm SE1 will involve $\sum_{i=1}^{S_1}\min\{N^{(i)}, K^{(i)}\}$ MVMs with matrix $\mathbf{B}$ and the integer S_1 will satisfy

$$\frac{r_{\max}\min\left\{1 - \sum_{j=0}^{N^{(S_1)}} P_j(\Lambda t_n), 2(M_{K^{(S_1)}} - m_{K^{(S_1)}})\right\}}{\min_{1\le i\le n}\widehat{\mathrm{ETRR}}^{(S_1)}(t_i)} \le \varepsilon\,.$$

For medium-sized and large MRMs, we can expect those MVMs to dominate the run-time computational cost of Algorithm SE1. However, by Eq. 11, Ineq. 13, and Eq. 14, assuming $\widehat{\mathrm{ETRR}}^{(S_1)}(t_n) \approx \widehat{\mathrm{ETRR}}(t_n, K_1)$, the truncation parameter K_1 of Algorithm 1 will satisfy

$$\begin{aligned}\frac{r_{\max}(M_{K_1} - m_{K_1})\left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t_n)\right)}{\widehat{\mathrm{ETRR}}^{(S_1)}(t_n)} &\approx \frac{r_{\max}(M_{K_1} - m_{K_1})\left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t_n)\right)}{\widehat{\mathrm{ETRR}}(t_n, K_1)}\\ &\le \frac{(M_{K_1} - m_{K_1})\left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t_n)\right)}{\sum_{j=0}^{K_1} v_j P_j(\Lambda t_n)}\\ &\le 2\varepsilon + (m_{K_1} + M_{K_1})\frac{1 - \sum_{j=0}^{K_1} P_j(\Lambda t_n)}{\sum_{j=0}^{K_1} v_j P_j(\Lambda t_n)}\varepsilon'\,,\end{aligned}$$

which is $> 2\varepsilon$. Therefore, taking into account that

$$(M_k - m_k)\left(1 - \sum_{j=0}^{k} P_j(\Lambda t_n)\right) \le \min\left\{\left(1 - \sum_{j=0}^{k} P_j(\Lambda t_n)\right), M_k - m_k\right\},$$

that the left-hand side of the above inequality is decreasing on k, and that $1/\min_{1\le i\le n}\widehat{\mathrm{ETRR}}^{(S_1)}(t_i) \ge 1/\widehat{\mathrm{ETRR}}^{(S_1)}(t_n)$, we can expect $K_1 \le \min\{N^{(S_1)}, K^{(S_1)}\} \le \sum_{i=1}^{S_1}\min\{N^{(i)}, K^{(i)}\}$. In addition, given the way Algorithm SE1 works, we can expect $S_1 = 1$ if $\min_{1\le i\le n}\mathrm{ETRR}(t_i)$ is not smaller than $(r_{\min} + r_{\max})/2$ and otherwise can expect $S_1 \ge 2$. Therefore, for medium-sized and large MRMs, we can expect the run-time computational cost of Algorithm 1 to be lower than that of Algorithm SE1, and the smaller $\min_{1\le i\le n}\mathrm{ETRR}(t_i)$ than $(r_{\min}+r_{\max})/2$, the larger we can expect the difference in run-time computational cost to be.

Let S_2 denote the number of times the second algorithm developed in Sericola (1999) is invoked in Algorithm SE2. Invocation j, $1 \le j \le S_2$, of that algorithm involves a number of MVMs with matrix $\mathbf{B}$ equal to $\min\{N^{(j)}, K^{(j)}\}$, where $N^{(j)}$ and $K^{(j)}$ are given by, respectively, Eqs. 41 and 42. Therefore, Algorithm SE2 will involve $\sum_{j=1}^{S_2} \min\{N^{(j)}, K^{(j)}\}$ MVMs with matrix **B** and the integer S_2 will satisfy

$$\frac{r_{\max} \min\left\{1 - \sum_{j=0}^{N^{(S_2)}} P_j(\Lambda t_n), 2(M_{K^{(S_2)}} - m_{K^{(S_2)}})\right\}}{\min_{1\le i\le n} \widehat{\mathrm{EARR}}^{(S_2)}(t_i)} \le \varepsilon\,.$$

For medium-sized and large MRMs, we can expect those MVMs to dominate the run-time computational cost of Algorithm SE2. On the other hand, by Eq. 34, Ineq. 35, and Eqs. 36 and 22, assuming $\widehat{\mathrm{EARR}}^{(S_2)}(t_n) \approx \widehat{\mathrm{EARR}}(t_i, K_2)$, the truncation parameter K_2 in Algorithm 2 will satisfy

$$\begin{aligned}
&\frac{r_{\max}(M_{K_2} - m_{K_2})\left(P_{K_2+1}(\Lambda t) + \frac{\Lambda t-(K_2+1)}{\Lambda t}(1-\sum_{j=0}^{K_2+1} P_j(\Lambda t_n))\right)}{\widehat{\mathrm{EARR}}^{(S_2)}(t_n)} \\
&\quad\approx \frac{r_{\max}(M_{K_2} - m_{K_2})\left(P_{K_2+1}(\Lambda t) + \frac{\Lambda t-(K_2+1)}{\Lambda t}(1-\sum_{j=0}^{K_2+1} P_j(\Lambda t_n))\right)}{\widehat{\mathrm{EARR}}(t_n, K_2)} \\
&\quad\le \frac{r_{\max}(M_{K_2} - m_{K_2})\left(P_{K_2+1}(\Lambda t) + \frac{\Lambda t-(K_2+1)}{\Lambda t}(1-\sum_{j=0}^{K_2+1} P_j(\Lambda t_n))\right)}{r_{\max}\sum_{j=0}^{K_2} w_j P_j(\Lambda t_n)} \\
&\quad\le 2\varepsilon + \frac{P_{K_2+1}(\Lambda t) + \frac{\Lambda t-(K_2+1)}{\Lambda t}(1-\sum_{j=0}^{K_2+1} P_j(\Lambda t_n))}{\sum_{j=0}^{K_2} w_j P_j(\Lambda t_n)}(m_{K_2}+M_{K_2})\varepsilon' \\
&\quad= 2\varepsilon + \frac{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t)}{\sum_{j=0}^{K_2} w_j P_j(\Lambda t_n)}(m_{K_2} + M_{K_2})\varepsilon'\,.
\end{aligned}$$

which is $> 2\varepsilon$. Therefore, taking into account that, by Eq. 22,

$$\begin{aligned}
&(M_k - m_k)\left(P_{k+1}(\Lambda t) + \frac{\Lambda t - (k+1)}{\Lambda t}\left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t)\right)\right) \\
&= (M_k - m_k)\left(1 - \sum_{j=0}^{k} P_j(\Lambda t_n) - \frac{k+1}{\Lambda t_n}\left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t_n)\right)\right) \\
&\le (M_k - m_k)\left(1 - \sum_{j=0}^{k} P_j(\Lambda t_n)\right) \\
&\le \min\left\{\left(1 - \sum_{j=0}^{k} P_j(\Lambda t_n)\right), M_k - m_k)\right\}\,,
\end{aligned}$$

that, again by Eq. 22, the left-hand side of the above inequality is decreasing on k, and that $1/\min_{1\le i\le n} \widehat{\mathrm{EARR}}^{(S_2)}(t_i) \ge 1/\widehat{\mathrm{EARR}}^{(S_2)}(t_n)$, we can expect $K_2 \le \min\{N^{(S_2)}, K^{(S_2)}\} \le \sum_{j=1}^{S_2} \min\{N^{(j)}, K^{(j)}\}$. Further, given the way Algorithm SE2 works, we can expect $S_2 = 1$ if $\min_{1\le i\le n} \mathrm{EARR}(t_i)$ is not smaller than $(r_{\min} + r_{\max})/2$ and otherwise can expect $S_2 \ge 2$. Therefore, we also conclude that, for medium-sized and large MRMs, we can expect the run-time computational cost of Algorithm 2 to be lower than that of Algorithm SE2 and that the smaller $\min_{1\le i\le n} \mathrm{EARR}(t_i)$ than $(r_{\min}+r_{\max})/2$, the larger we can expect the difference in run-time computational cost to be.

5 Numerical Experiments

In this section, we will illustrate the performances of Algorithms 1 and 2. We will also compare the performance of Algorithm 1 with the performances of Algorithms SC1 and SE1 and will compare the performance of Algorithm 2 with the performances of Algorithms SC2 and SE2.

5.1 Examples

We will use two MRMs. The first one is taken from Carrasco (2003b) and corresponds to a RAID 5 storage system (Chen et al. 1994) with the architecture shown in Fig. 1. The system comprises 40×5 disks, 5 controllers, 3 hot spare disks, and 1 hot spare controller. The disks are organized into 40 parity groups with 5 disks each. Each controller controls a string of 40 disks. A disk is said to be unavailable if it has failed, or the controller of the string the disk belongs to has failed, or the data in the disk is out of date. The system is in a failed state if there is any parity group in which two or more disks are unavailable. The data of a non-failed disk becomes out of date if it is a disk that has just replaced a failed one or it belongs to a string of disks whose controller was failed and has just been replaced. Out-of-date disks become up-to-date after a reconstruction process that proceeds at a rate of $1\,\mathrm{h}^{-1}$. That process has a success probability 0.999 and can take place only when the system is in a non-failed state. All disks of a parity group involved in a reconstruction process fail at a rate of $2 \times 10^{-5}\,\mathrm{h}^{-1}$. Disks not involved in a reconstruction process fail at a rate of $1 \times 10^{-5}\,\mathrm{h}^{-1}$. Controllers fail at a rate of $5 \times 10^{-5}\,\mathrm{h}^{-1}$. There is one repair person that, if hot spares are available, replaces failed disks and controllers at a rate of $4\,\mathrm{h}^{-1}$, with priority given to controllers. There is an unlimited number of repair persons that replace used hot spares and failed disks and controllers when no hot spares are available at a rate of $0.25\,\mathrm{h}^{-1}$. When the system is in a failed state, no components fail or are repaired and the only repair action is one that brings the system to its fully operational state, with all disks in the parity groups available and all hot spares available, at a rate of $0.25\,\mathrm{h}^{-1}$. The initial probability is one for the fully operational state and is zero for the remaining states. As in Carrasco (2003b), we assume that if unavailable disks do not belong to the same string, when one of them becomes available the remaining unavailable disks still belong to different strings whenever their number is ≥ 2. The CTMC has 14081 states and 94405 transition rates. The reward rates are $r_i = 1$ for the nonfailed states and $r_i = 0$ for the failed states. With those reward rates, $\mathrm{ETRR}(t)$ is the availability of the system at time t (probability that the system

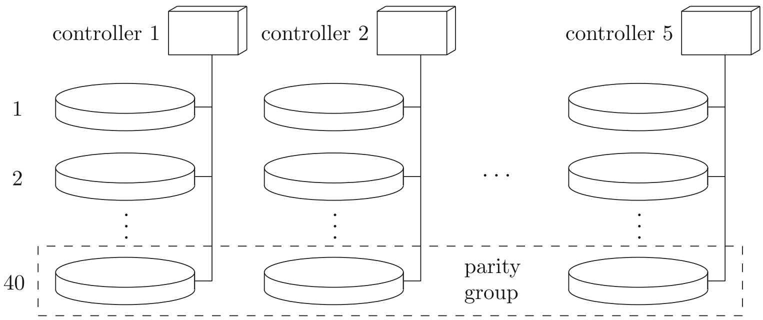


Fig. 1 Architecture of the RAID 5 storage system

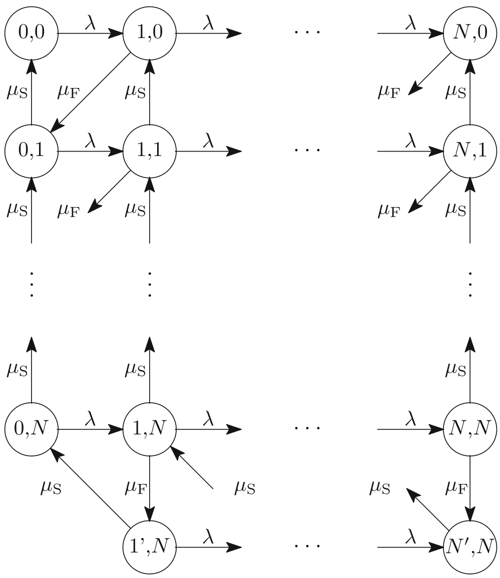
is not failed at time t) and EARR(t) is the expected interval availability of the system in the time interval [0, t] (expected fraction of the time interval in which the system is not failed). In addition, $r_{\min} = 0_x$, and $r_{\max} = 1$.

The second MRM, adapted from Carrasco (2015), corresponds to two FIFO queues working in tandem. Each queue has a buffer with capacity for $N = 100$ tasks. Tasks arrive on the first queue with rate $\lambda = 2\,\mathrm{h}^{-1}$. When a task of the first queue is served, it is delivered to the second queue unless it is full, in which case the task is blocked until there is room for it in the second queue. The service rates are $\mu_\mathrm{F} = 2.2\,\mathrm{h}^{-1}$ for the first queue and $\mu_\mathrm{S} = 2.5\,\mathrm{h}^{-1}$ for the second one. The initial probability is one for the state in which there are no tasks in the system and is zero for the remaining states. The CTMC has 10301 states and 30499 transition rates. Its state transition diagram is shown in Fig. 2, where the states in which there are i tasks in the first queue and j tasks in the second one and no task is blocked are labeled "i, j", and the states in which there are i tasks in the first queue, N tasks in the second queue, and a task of the first queue is blocked are labeled "i',N". The reward rates are the number of tasks in the system. With those reward rates, ETRR(t) is the expected number of tasks in the system at time t and EARR(t) is the expected average number of tasks in the system in the time interval [0, t]. Besides, $r_{\min} = 0_x$, and $r_{\max} = 200$.

5.2 Results

All algorithms were implemented using the C programming language, with all floating-point computations performed using the IEEE754-1985 (IEEE754 1985) double format, and were compiled using the standard GNU compiler collection C-compiler (Stallman et al. 2012) with the `O2` optimization option. The input parameter δ of the proposed algorithms was set to $10^3 \times 2^{-52}$. In all cases, Poisson probabilities were computed using the method described in (Knüsel 1986, pp. 1028–1029) (see also Abramowitz and Stegun 1964) and Λ was set to $\Lambda = \theta \times \max_{i\in\mathcal{S}}|a_{i,i}|$ with $\theta = 1.001 > 1$ to ensure that Eq. 2 holds. This latter setting resulted in $\Lambda \approx 43.80\,\mathrm{h}^{-1}$ for the RAID 5 storage system MRM and $\Lambda \approx 6.707\,\mathrm{h}^{-1}$

Fig. 2 State transition diagram of the queueing MRM



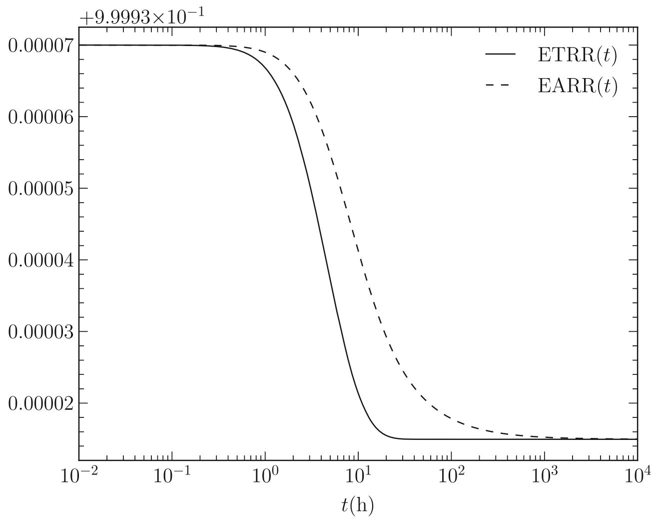


Fig. 3 ETRR(t) and EARR(t) for the RAID 5 storage system MRM

for the queueing MRM. All results were obtained on a workstation equipped with a four-core Intel i7-2630QM 2.00 GHz processor with 4 GB of RAM memory, using only one core.

In Figs. 3 and 4, we plot ETRR(t) and EARR(t) for the RAID 5 storage system MRM and the queueing MRM, respectively, obtained executing the proposed algorithms for 300 time points equally spaced in a logarithmic scale with a relative error tolerance $\varepsilon = 10^{-10}$. We can observe that, for the first example, ETRR(t) "reaches" the stationary value very soon, at about $t = 100$ h, that EARR(t) "reaches" the stationary value somewhat later, and that both ETRR(t) and EARR(t) are larger than $(r_{\min} + r_{\max})/2 = 0.5$. For the queueing

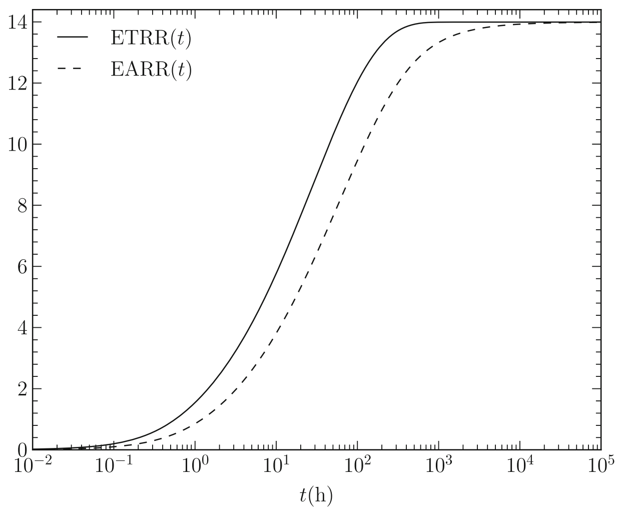


Fig. 4 ETRR(t) and EARR(t) for the queueing MRM

MRM, we observe, both ETRR(t) and EARR(t) "reach" the stationary value later, between $t = 1000$ h and $t = 10000$ h, and are significantly smaller than $(r_{\min} + r_{\max})/2 = 100$.

We will compare the proposed algorithms with the alternatives from a triple perspective: run-time computational cost measured in terms of CPU time, relative accuracy, and run-time computational cost in relation to relative accuracy. To carry out those comparisons, we executed each algorithm for both MRMs with $n = 1$, $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$, and $\varepsilon = 10^{-4}, 10^{-5}, \ldots, 10^{-12}$. The reference values for ETRR(t) and EARR(t), $t = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$, with which to compute the algorithms' accuracy were obtained using the implementation of the randomization method described in (Suñé and Carrasco 2005, Section 1), computing Poisson probabilities using a variant of the algorithm described in Fox and Glynn (1988), which is numerically very stable, and performing all floating-point computations using the IEEE 754-2008 binary128 format (IEEE754-2008 2008) emulated with the MPFR library (Fousse et al. 2007). (The binary128 format gives around 34 decimal digits precision as opposed to the approximately 16 decimal digits precision given by the double format.)

The numbers of MVMs with matrix **B** required by the proposed algorithms were always smaller than the numbers of MVMs with matrix **B** required by the alternatives and, consequently, the CPU times of the proposed algorithms were almost always lower than those of the alternatives. In addition, the larger t_n, the larger were the differences in terms of CPU time between Algorithm 1 and Algorithm SC1 and between Algorithm 2 and Algorithm SC2. As an illustration, in Tables 1, 2, 3 and 4 we give the number of MVMs with matrix **B** and the CPU time for each of the algorithms, for $\varepsilon = 10^{-6},\ 10^{-10}$.

The tables also illustrate the fact that, as commented in Section 4, how the run-time computational cost of Algorithm 1 compares with that of Algorithm SE1 depends on whether $\min_{1\le i\le n} \text{ETRR}(t_i) \ge (r_{\min} + r_{\max})/2$, and how the run-time computational cost of Algorithm 2 compares with that of Algorithm SE2 depends on whether $\min_{1\le i\le n} \text{EARR}(t_i) \ge$

Table 1 RAID 5 storage system MRM: numbers of MVMs with matrix **B** (top) and CPU times in seconds (bottom) required by Algorithms 1, SC1, and SE1 to compute ETRR(t_n), $n = 1$

	$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-10}$		
t_n (h)	Alg. 1	Alg. SC1	Alg. SE1	Alg. 1	Alg. SC1	Alg. SE1
5	285 (359)	297	295	313 (377)	323	321
	4.80×10^{-2}	5.20×10^{-2}	4.80×10^{-2}	5.20×10^{-2}	5.60×10^{-2}	5.20×10^{-2}
10	523 (628)	547	544	563 (651)	582	580
	8.40×10^{-2}	9.20×10^{-2}	8.80×10^{-2}	8.80×10^{-2}	1.00×10^{-1}	9.60×10^{-2}
100	2293 (NC)	4712	2656	3902 (NC)	4817	4265
	3.72×10^{-1}	8.12×10^{-1}	4.32×10^{-1}	6.36×10^{-1}	8.28×10^{-1}	6.92×10^{-1}
1000	2293 (NC)	44810	2656	3902 (NC)	45136	4265
	3.68×10^{-1}	7.74	4.32×10^{-1}	6.36×10^{-1}	7.77	6.96×10^{-1}
10000	2293 (NC)	440871	2656	3902 (NC)	441895	4265
	3.72×10^{-1}	7.55×10^{1}	4.44×10^{-1}	6.32×10^{-1}	7.63×10^{1}	7.04×10^{-1}
100000	2293 (NC)	4385937	2656	3902 (NC)	4389171	4265
	3.76×10^{-1}	7.57×10^{2}	5.24×10^{-1}	6.28×10^{-1}	7.52×10^{2}	7.88×10^{-1}

(For Algorithm 1, next to the number of MVMs we give between parenthesis the value of the truncation parameter R_1, with "NC" standing for "not computed")

Table 2 RAID 5 storage system MRM: numbers of MVMs with matrix **B** (top) and CPU times in seconds (bottom) required by Algorithms 2, SC2, and SE2 to compute EARR(t_n), $n = 1$

	$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-10}$		
t_n (h)	Alg. 2	Alg. SC2	Alg. SE2	Alg. 2	Alg. SC2	Alg. SE2
5	269 (352, 357)	297	295	300 (369, 374)	323	321
	4.40×10^{-2}	5.20×10^{-2}	4.80×10^{-2}	5.20×10^{-2}	5.60×10^{-2}	5.20×10^{-2}
10	498 (617, 624)	547	544	544 (640, 647)	582	580
	8.00×10^{-2}	9.20×10^{-2}	8.80×10^{-2}	8.80×10^{-2}	1.00×10^{-1}	9.60×10^{-2}
100	2293 (NC, NC)	4712	2656	3902 (NC, NC)	4817	4265
	3.72×10^{-1}	8.08×10^{-1}	4.32×10^{-1}	6.36×10^{-1}	8.32×10^{-1}	6.92×10^{-1}
1000	2293 (NC, NC)	44810	2656	3902 (NC, NC)	45136	4265
	3.68×10^{-1}	7.80	4.36×10^{-1}	6.36×10^{-1}	7.71	6.96×10^{-1}
10000	2293 (NC, NC)	440871	2656	3902 (NC, NC)	441895	4265
	3.72×10^{-1}	7.66×10^{1}	4.44×10^{-1}	6.32×10^{-1}	7.60×10^{1}	7.04×10^{-1}
100000	2293 (NC, NC)	4385937	2656	3902 (NC, NC)	4389171	4265
	3.72×10^{-1}	7.56×10^{2}	5.24×10^{-1}	6.28×10^{-1}	7.58×10^{2}	7.88×10^{-1}

(For Algorithm 2, next to the number of MVMs we give between parenthesis the value of the truncation parameters R_2 and R_3, with “NC” standing for “not computed”)

$(r_{\min} + r_{\max})/2$. Thus, for the first MRM, for which both ETRR(t_n) and EARR(t_n), $t_n = 5\,\mathrm{h}, 10\,\mathrm{h}, \ldots, 10^5\,\mathrm{h}$, are larger than $(r_{\min} + r_{\max})/2 = 0.5$, Algorithms SE1 and SE2 always terminated in one iteration and, as a consequence, Algorithm 1 was only slightly faster than Algorithm SE1 and Algorithm 2 was only slightly faster than Algorithm SE2. On the contrary, for the second MRM, for which both ETRR(t_n) and EARR(t_n),

Table 3 Queueing MRM: numbers of MVMs with matrix **B** (top) and CPU times in seconds (bottom) required by Algorithms 1, SC1, and SE1 to compute ETRR(t_n), $n = 1$

	$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-10}$		
t_n (h)	Alg. 1	Alg. SC1	Alg. SE1	Alg. 1	Alg. SC1	Alg. SE1
5	69 (100)	72	135	80 (108)	82	158
	4.00×10^{-3}	4.00×10^{-3}	8.00×10^{-3}	8.00×10^{-3}	4.00×10^{-3}	8.00×10^{-3}
10	115 (156)	118	227	130 (167)	133	257
	8.00×10^{-3}	8.00×10^{-3}	1.60×10^{-2}	8.00×10^{-3}	1.20×10^{-2}	2.00×10^{-2}
100	806 (920)	818	1613	848 (947)	858	1699
	5.60×10^{-2}	5.60×10^{-2}	1.12×10^{-1}	6.00×10^{-2}	6.00×10^{-2}	1.20×10^{-1}
1000	7009 (7404)	7156	14253	7166 (7490)	7277	14511
	4.92×10^{-1}	5.00×10^{-1}	1.01	5.08×10^{-1}	5.04×10^{-1}	1.02
10000	15912 (NC)	68432	49949	25001 (NC)	68808	77212
	1.12	4.78	3.52	1.77	4.81	5.43
100000	15912 (NC)	674519	49949	25001 (NC)	675702	77212
	1.13	4.66×10^{1}	3.57	1.76	4.72×10^{1}	5.50

(For Algorithm 1, next to the number of MVMs we give between parenthesis the value of the truncation parameter R_1, with “NC” standing for “not computed”)

Table 4 Queueing MRM: numbers of MVMs with matrix **B** (top) and CPU times in seconds (bottom) required by Algorithms 2, SC2, and SE2 to compute EARR(t_n), $n = 1$

	$\varepsilon = 10^{-6}$			$\varepsilon = 10^{-10}$		
t_n (h)	Alg. 2	Alg. SC2	Alg. SE2	Alg. 2	Alg. SC2	Alg. SE2
5	64 (98, 99)	72	135	76 (106, 108)	83	158
	4.00×10^{-3}	4.00×10^{-3}	1.20×10^{-2}	4.00×10^{-3}	4.00×10^{-3}	1.20×10^{-2}
10	108 (152, 155)	119	228	124 (163, 165)	133	257
	8.00×10^{-3}	8.00×10^{-3}	1.60×10^{-2}	1.20×10^{-2}	8.00×10^{-3}	2.00×10^{-2}
100	780 (906, 915)	819	1614	827 (934, 942)	859	1700
	5.60×10^{-2}	6.00×10^{-2}	1.12×10^{-1}	6.00×10^{-2}	6.00×10^{-2}	1.20×10^{-1}
1000	6872 (7351, 7379)	7157	14254	7072 (7437, 7465)	7278	14511
	4.88×10^{-1}	5.00×10^{-1}	1.01	4.96×10^{-1}	5.04×10^{-1}	1.02
10000	15912 (NC, NC)	68432	49958	25001 (NC, NC)	68809	77221
	1.12	4.73	3.55	1.76	4.77	5.48
100000	15912 (NC, NC)	674519	49950	25001 (NC, NC)	675702	77212
	1.12	4.65×10^{1}	3.56	1.77	4.70×10^{1}	5.52

(For Algorithm 2, next to the number of MVMs we give between parenthesis the value of the truncation parameters R_2 and R_3, with "NC" standing for "not computed")

$t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$, are much smaller than $(r_{\min}+r_{\max})/2 = 100$, Algorithms SE1 and SE2 always required between two and three iterations and were therefore noticeably slower than, respectively, Algorithms 1 and 2. Also given in the tables are the values taken by the truncation parameters R_1, R_2, and R_3. As we can see, for the RAID 5 storage system MRM, the parameters were computed only for $t_n = 5, 10\,\text{h}$ and their values are larger than the numbers of MVMs with matrix **B** required by the alternatives, and for the queueing MRM, the parameters were computed for $t_n = 5, 10, 100, 1\,000\,\text{h}$ and their values are moderately larger than the numbers of MVMs with matrix **B** required by Algorithms SC1 and SC2 and smaller than the numbers of MVMs with matrix **B** required by Algorithms SE1 and SE2. However, computing the truncation parameter R_1 defined by Eq. 8 essentially amounts to obtaining R_1+1-K_1 additional Poisson probabilities and computing the truncation parameters R_2 and R_3 defined by, respectively, Eqs. 28 and 30, essentially amounts to obtaining $\max\{R_2, R_3+1\} - K_2$ additional Poisson probabilities. Therefore, for medium-sized and large MRMs, assuming, quite reasonably, that the run-time computational cost of performing one MVM with matrix **B** will be substantially higher than the run-time computational cost of computing one Poisson probability, we expect the computation of those parameters to have a very small impact on the run-time computational cost of the proposed algorithms. That this is certainly the case for the two MRMs we are considering can be easily realized by comparing the CPU times required by the proposed algorithms with those required by Algorithms SC1 and SC2 in the cases in which the truncation parameters were computed.

Since the proposed algorithms and the alternatives all control the approximation error relative to the computed estimate, we cannot expect Algorithms 1, SC1, and SE1 to yield the same approximation for ETRR(t) within the relative error tolerance nor can expect Algorithms 2, SC2, and SE2 to yield the same approximation for EARR(t) within the relative error tolerance. Instead, what must happen is that the error of each estimate relative to the estimate itself is nonlarger than ε. To compare, in terms of relative accuracy, Algorithm 1 with Algorithms SC1 and SE1, and Algorithm 2 with Algorithms SC2 and SE2,

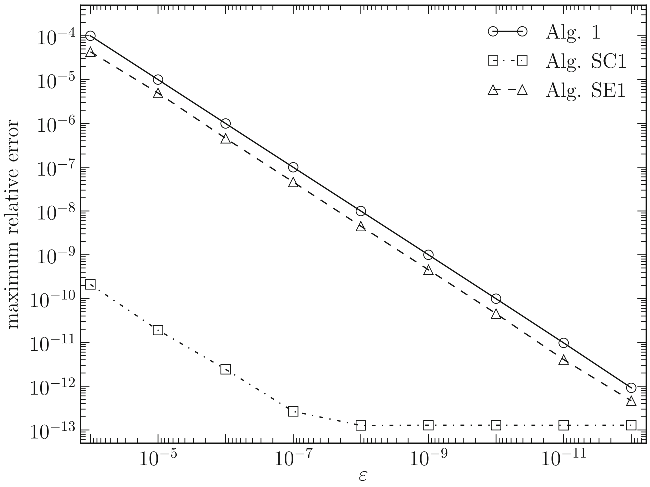


Fig. 5 RAID 5 storage system MRM: maximum over $t_n = 5\,\mathrm{h}, 10\,\mathrm{h}, \ldots, 10^5\,\mathrm{h}$ of the actual relative error in Algorithms 1, SC1, and SE1, as a function of ε

we computed the maximum over $t_n = 5\,\mathrm{h}, 10\,\mathrm{h}, \ldots, 10^5\,\mathrm{h}$ of the actual error relative to the computed estimate for each algorithm and each $\varepsilon = 10^{-4}, 10^{-5}, \ldots, 10^{-12}$. The results are shown in Figs. 5, 6 for the RAID 5 storage system MRM and in Figs. 7, 8 for the queueing MRM. As we can see, the control of the error in the proposed algorithms is very tight in the sense that the actual relative error is always very close to (but smaller than) the tolerance ε. We also see that, in almost all cases, the proposed algorithms are less accurate than the alternatives.

Finally, we note that, for $\varepsilon = 10^{-11}, 10^{-12}$, Algorithms SC1 and SC2 were unable to fulfill the accuracy requirement for the queueing MRM. We conjecture that this anomalous behavior is due to the cumulative effect of round-off errors.

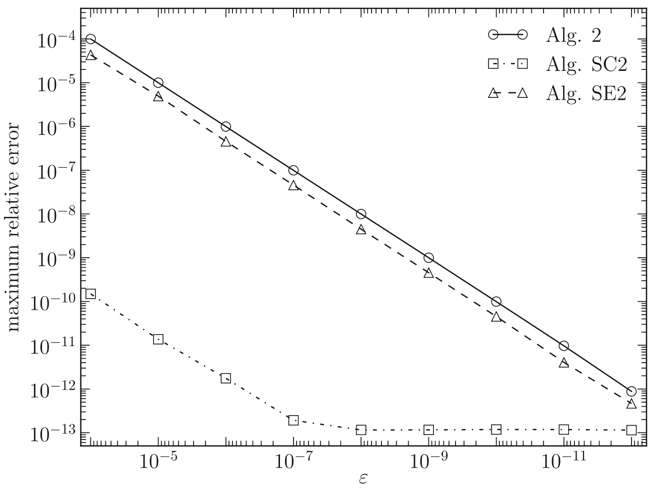


Fig. 6 RAID 5 storage system MRM: maximum over $t_n = 5\,\mathrm{h}, 10\,\mathrm{h}, \ldots, 10^5\,\mathrm{h}$ of the actual relative error in Algorithms 2, SC2, and SE2, as a function of ε

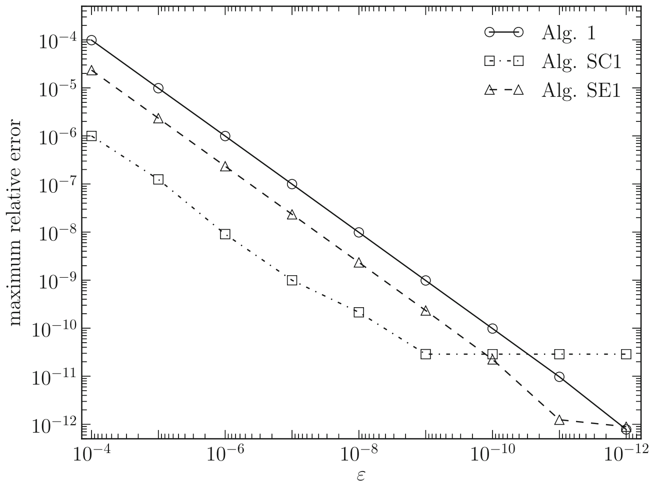


Fig. 7 Queueing MRM: maximum over $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$ of the actual relative error in Algorithms 1, SC1, and SE1, as a function of ε

The fact that the alternatives are more accurate than the proposed algorithms comes at the price of a higher run-time computational cost. Therefore, it would be fairer to compare the proposed algorithms with the alternatives from the perspective of relative accuracy in relation to run-time computational cost. To that end, in Figs. 9, 10, 11 and 12 we show the work-precision curves of the six algorithms. In each of these curves, the abscissa of the ith point, $i = 1, 2, \ldots, 9$, starting from the left, corresponds to the maximum over $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$ of the actual relative error when the algorithm was executed with a relative error tolerance $\varepsilon = 10^{-(3+i)}$, and the ordinate corresponds to the cumulative CPU time required by the algorithm for $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$. We can now see that,

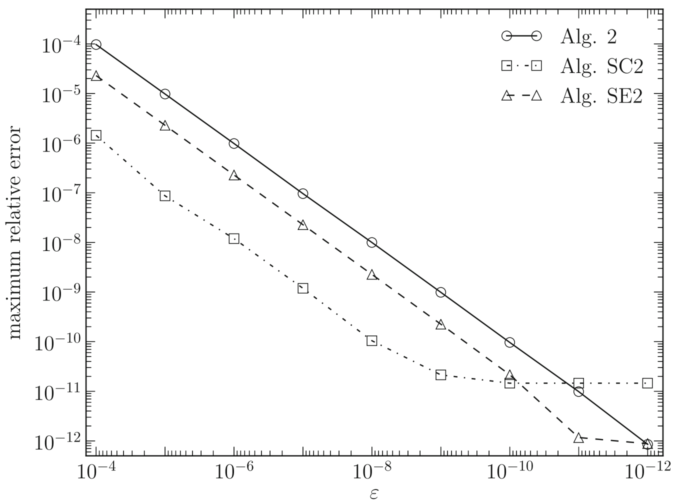


Fig. 8 Queueing MRM: maximum over $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$ of the actual relative error in Algorithms 2, SC2, and SE2, as a function of ε

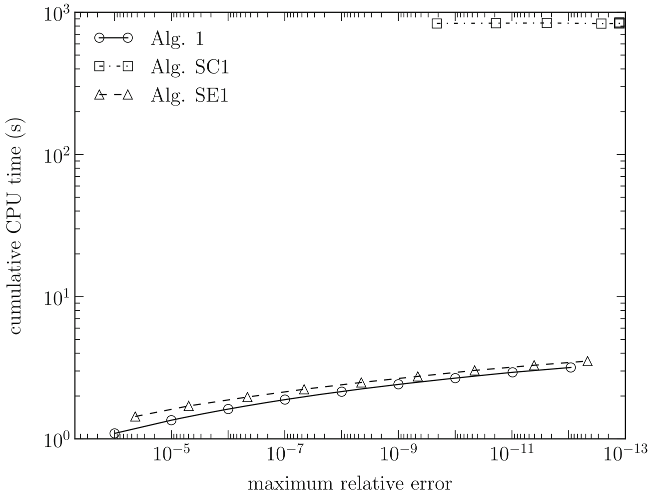


Fig. 9 RAID 5 storage system MRM: cumulative CPU time in seconds required to compute ETRR(t_n), $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$, as a function of the maximum over t_n of the actual relative error in Algorithms 1, SC1, and SE1 (In each case, the left-most symbol corresponds to $\varepsilon = 10^{-4}$)

in all cases, Algorithms 1 and 2 are much more efficient than Algorithms SC1 and SC2, respectively, in the sense of requiring a much smaller CPU time to achieve the same relative error, that Algorithm 1 is slightly more efficient than Algorithm SE1 for the RAID 5 storage system MRM and quite more efficient for the queueing MRM, and that, compared with Algorithm SE2, Algorithm 2 is slightly more efficient for the RAID 5 storage system MRM and quite more so for the queueing MRM.

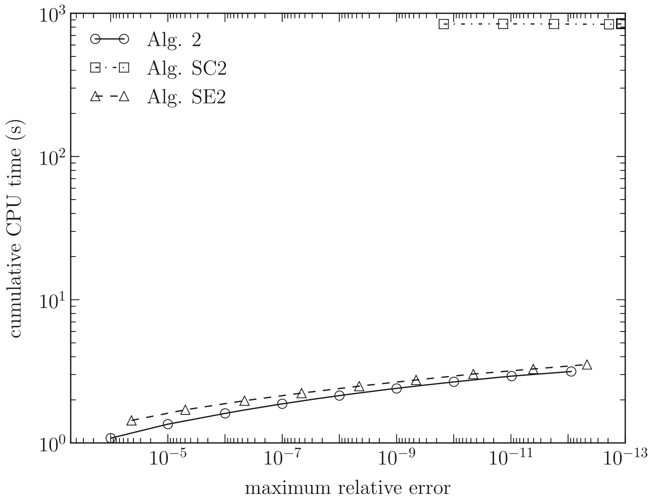


Fig. 10 RAID 5 storage system MRM: cumulative CPU time in seconds required to compute EARR(t_n), $t_n = 5\,\text{h}, 10\,\text{h}, \ldots, 10^5\,\text{h}$, as a function of the maximum over t_n of the actual relative error in Algorithms 2, SC2, and SE2 (In each case, the left-most symbol corresponds to $\varepsilon = 10^{-4}$)

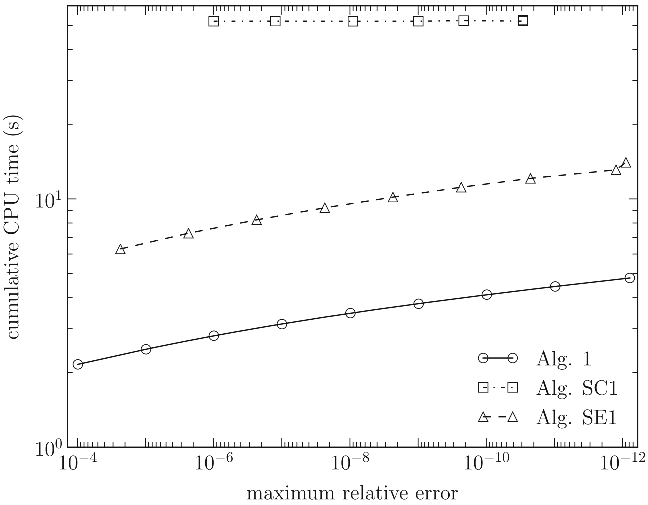


Fig. 11 Queueing MRM: cumulative CPU time in seconds required to compute ETRR(t_n), t_n = 5 h, 10 h, ..., 10^5 h, as a function of the maximum over t of the actual relative error in Algorithms 1, SC1, and SE1 (In each case, the left-most symbol corresponds to $\varepsilon = 10^{-4}$)

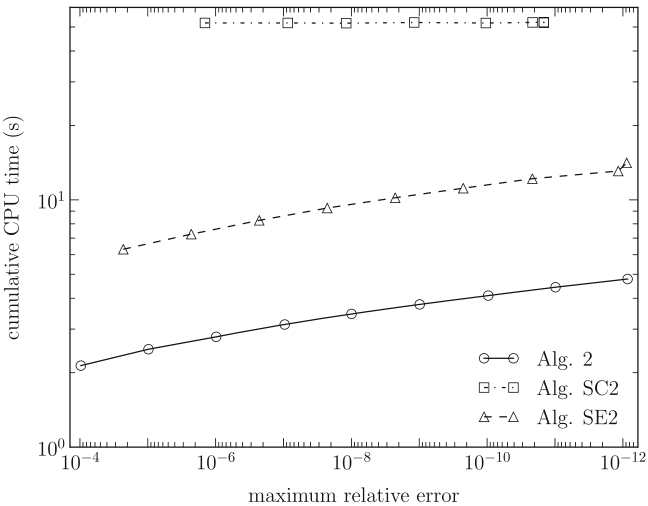


Fig. 12 Queueing MRM: cumulative CPU time in seconds required to compute EARR(t_n), t_n = 5 h, 10 h, ..., 10^5 h, as a function of the maximum over t_n of the actual relative error in Algorithms 2, SC2, and SE2 (In each case, the left-most symbol corresponds to $\varepsilon = 10^{-4}$)

6 Conclusions

In this paper, by combining in a novel way the randomization method with the stationarity detection technique proposed in Sericola (1999), we have developed two new algorithms for the computation of the expected reward rates of finite, irreducible MRMs, with control of the relative error. The first algorithm computes the expected transient reward rate and the second one computes the expected averaged reward rate. We have argued that the algorithms are numerically stable and that, for medium-sized and large MRMs, we can expect the run-time computational cost of the new algorithms to be lower than that of the variants of the randomization method developed in Suñé and Carrasco (2005), which allow to compute the expected reward rates with control of the relative error, and lower than the run-time computational cost of the approach that consists in using iteratively the algorithms developed in Sericola (1999), which allow to compute the expected reward rates with control of the absolute error. The performance of the algorithms has been illustrated numerically, showing that the algorithms can be not only faster but also substantially more efficient than the alternatives in the sense of being able to achieve the same accuracy with a much lower run-time computational cost.

Acknowledgments The author is indebted to Juan A. Carrasco for inspiring this investigation and to the referees, whose comments and suggestions have helped greatly to improve the paper.

Appendix A: Proofs

We will make use of Lemmas 1 and 2 given next.

Lemma 1 *Let* $0 < \lambda' \leq \lambda$, l, m, n, $0 \leq l \leq m \leq n$, *and* $f(k), g(k) \geq 0$, *with* $g(k)$ *uniformly upper bounded and strictly positive for some* k, $k \geq n$. *Then,*

$$\frac{\sum_{k=l}^{m} f(k) P_k(\lambda)}{\sum_{k=n}^{\infty} g(k) P_k(\lambda)} \leq \frac{\sum_{k=l}^{m} f(k) P_k(\lambda')}{\sum_{k=n}^{\infty} g(k) P_k(\lambda')} .$$

Proof Using Lemma 1 in Suñé and Carrasco (2005) with $w(k) = f(k)$, $i = l$, $j = m$, $\lambda_1 = \lambda$, $\lambda_2 = \lambda'$, and $x = \lambda/\lambda'$,

$$\left(\frac{\lambda'}{\lambda}\right)^m \mathrm{e}^{(\lambda-\lambda')} \sum_{k=l}^{m} f(k) P_k(\lambda) \leq \sum_{k=l}^{m} f(k) P_k(\lambda') . \tag{43}$$

Using again the lemma, now with $w(k) = g(k)$, $i = n$, $j = \infty$, $\lambda_1 = \lambda$, $\lambda_2 = \lambda'$, and $x = \lambda/\lambda'$,

$$\sum_{k=n}^{\infty} g(k) P_k(\lambda') \leq \left(\frac{\lambda'}{\lambda}\right)^n \mathrm{e}^{(\lambda-\lambda')} \sum_{k=n}^{\infty} g(k) P_k(\lambda) . \tag{44}$$

Combining Ineqs. 43, 44, recalling that, by assumption, $g(k) > 0$ for some k, $k \geq n$, and noting that, for $n \geq m$, $(\lambda'/\lambda)^{n-m} \leq 1$,

$$\begin{aligned} \frac{\sum_{k=l}^{m} f(k) P_k(\lambda)}{\sum_{k=n}^{\infty} g(k) P_k(\lambda)} &\leq \frac{\sum_{k=l}^{n} f(k) P_k(\lambda')}{\left(\frac{\lambda'}{\lambda}\right)^m \mathrm{e}^{(\lambda-\lambda')}} \frac{\left(\frac{\lambda'}{\lambda}\right)^n \mathrm{e}^{(\lambda-\lambda')}}{\sum_{k=n}^{\infty} g(k) P_k(\lambda')} \\ &= \left(\frac{\lambda'}{\lambda}\right)^{n-m} \frac{\sum_{k=l}^{m} f(k) P_k(\lambda')}{\sum_{k=n+1}^{\infty} g(k) P_k(\lambda')} \\ &\leq \frac{\sum_{k=l}^{m} f(k) P_k(\lambda')}{\sum_{k=n+1}^{\infty} g(k) P_k(\lambda')} . \end{aligned}$$

□

Lemma 2 *Assume* $\lambda > 0$, $n \geq 0$, *and* $r \geq n$. *If* $r > \lambda - 2$, $m \geq 1$, *and* $m' \geq 1$, *then*

$$\sum_{j=r}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda) \leq \left(1-\left(\frac{\lambda}{r+2}\right)^m\right)^{-1}\left(\sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda)\right.$$
$$\left.+\left(\frac{\lambda}{r+2}\right)^m \frac{m}{\lambda}\left(1-\left(\frac{\lambda}{r+2}\right)^{m'}\right)^{-1}\sum_{j=r+1}^{r+m'}P_j(\lambda)\right).$$

Proof Using that, for $j \geq r$,

$$P_{j+m+1}(\lambda) = \lambda^{m+1}\frac{j!}{(j+m+1)!}P_j(\lambda) = \lambda^{m+1}\frac{1}{j+1}\prod_{i=2}^{m+1}\frac{1}{j+i}P_j(\lambda)$$
$$\leq \lambda^{m+1}\frac{1}{j+1}\left(\frac{1}{r+2}\right)^m P_j(\lambda),$$

we can write

$$\sum_{j=r}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda) = \sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda) + \sum_{j=r+m}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda)$$
$$= \sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda) + \frac{1}{\lambda}\sum_{j=r+m}^{\infty}(j+1-n)P_{j+1}(\lambda)$$
$$= \sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda) + \frac{1}{\lambda}\sum_{j=r}^{\infty}(j+m+1-n)P_{j+m+1}(\lambda)$$
$$\leq \sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda)+\frac{1}{\lambda}\sum_{j=r}^{\infty}(j+m+1-n)\lambda^{m+1}\frac{1}{j+1}\left(\frac{1}{r+2}\right)^m P_j(\lambda)$$
$$= \sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda) + \left(\frac{\lambda}{r+2}\right)^m\sum_{j=r}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda)$$
$$+\left(\frac{\lambda}{r+2}\right)^m\sum_{j=r}^{\infty}\frac{m}{j+1}P_j(\lambda)$$
$$= \sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda) + \left(\frac{\lambda}{r+2}\right)^m\sum_{j=r}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda)$$
$$+\left(\frac{\lambda}{r+2}\right)^m\frac{m}{\lambda}\sum_{j=r+1}^{\infty}P_j(\lambda).$$

Since $r > \lambda - 2$, we can invoke Proposition 1 in Glynn (1987) with n replaced by $r+1$ and m replaced by m' to bound $\sum_{j=r+1}^{\infty}P_j(\lambda)$ from above, obtaining

$$\sum_{j=r+1}^{\infty}P_j(\lambda) \leq \left(1-\left(\frac{\lambda}{r+2}\right)^{m'}\right)^{-1}\sum_{j=r+1}^{r+m'}P_j(\lambda).$$

Therefore,

$$\sum_{j=r}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda)\leq\sum_{j=r}^{r+m-1}\left(1-\frac{n}{j+1}\right)P_j(\lambda)+\left(\frac{\lambda}{r+2}\right)^m\sum_{j=r}^{\infty}\left(1-\frac{n}{j+1}\right)P_j(\lambda)$$
$$+\left(\frac{\lambda}{r+2}\right)^m\frac{m}{\lambda}\left(1-\left(\frac{\lambda}{r+2}\right)^{m'}\right)^{-1}\sum_{j=r+1}^{r+m'}P_j(\lambda),$$

and the result follows by solving for $\sum_{j=r}^{\infty}(1-n/(j+1))P_j(\lambda)$. □

Proof of Proposition 1 That the truncation parameter R_1 given by Eq. 9 is finite follows from

$$\lim_{r\to\infty}\frac{\frac{1}{r+3-\Lambda t}\frac{r+3}{r+2}\left(r+2+\frac{\Lambda t}{r+3-\Lambda t}\right)P_{r+1}(\Lambda t)}{\sum_{j=k+1}^{r}P_j(\Lambda t)}=0$$

and the fact that, as assumed, $\delta>0$. With regard to Ineq. (10), it trivially holds in the case $\sum_{j=0}^{k}P_j(\Lambda t)\leq 0.9$. In the case $\sum_{j=0}^{k}P_j(\Lambda t)>0.9$, we clearly have, by Eq. 8,

$$0\leq\frac{\sum_{j=k+1}^{\infty}P_j(\Lambda t)-\sum_{j=k+1}^{R_1}P_j(\Lambda t)}{\sum_{j=k+1}^{\infty}P_j(\Lambda t)}=\frac{1-\sum_{j=0}^{k}P_j(\Lambda t)-[1-\sum_{j=0}^{k}P_j(\Lambda t)]^{\mathrm{lb}}}{1-\sum_{j=0}^{k}P_j(\Lambda t)}.$$

Besides, since the median of a Poisson distribution with parameter Λt is nonsmaller than $\Lambda t-\log 2$ (Choi 1994), we also have $k\geq\Lambda t-\log 2>\Lambda t-2$. Therefore, by Eq. 9, we have $R_1\geq k+1>\Lambda t-1$, which implies $R_1+1>\Lambda t-2$, and can then bound $\sum_{j=R_1+1}^{\infty}P_j(\Lambda t)$ from above by invoking Lemma 2 with $r=R_1+1$, $n=0$, $\lambda=\Lambda t$, $m=1$, and $m'=1$. The result is:

$$\sum_{j=R_1+1}^{\infty}P_j(\Lambda t)\leq\left(1-\frac{\Lambda t}{R_1+3}\right)^{-1}\left(P_{R_1+1}(\Lambda t)+\frac{\Lambda t}{R_1+3}\frac{1}{\Lambda t}\left(1-\frac{\Lambda t}{R_1+3}\right)^{-1}P_{R_1+2}(\Lambda t)\right).\tag{45}$$

Using then Eq. 8, Ineq. 45, $P_{R_1+2}(\Lambda t)=(\Lambda t/(R_1+2))P_{R_1+1}(\Lambda t)$, and Eq. 9,

$$\begin{aligned}\frac{1-\sum_{j=0}^{k}P_j(\Lambda t)-[1-\sum_{j=0}^{k}P_j(\Lambda t)]^{\mathrm{lb}}}{1-\sum_{j=0}^{k}P_j(\Lambda t)}&=\frac{\sum_{j=k+1}^{\infty}P_j(\Lambda t)-\sum_{j=k+1}^{R_1}P_j(\Lambda t)}{\sum_{j=k+1}^{\infty}P_j(\Lambda t)}\\&=\frac{\sum_{j=R_1+1}^{\infty}P_j(\Lambda t)}{\sum_{j=k+1}^{\infty}P_j(\Lambda t)}\\&<\frac{\sum_{j=R_1+1}^{\infty}P_j(\Lambda t)}{\sum_{j=k+1}^{R_1}P_j(\Lambda t)}\\&\leq\frac{\left(1-\frac{\Lambda t}{R_1+3}\right)^{-1}\left(P_{R_1+1}(\Lambda t)+\frac{\Lambda t}{R_1+3}\frac{1}{\Lambda t}\left(1-\frac{\Lambda t}{R_1+3}\right)^{-1}P_{R_1+2}(\Lambda t)\right)}{\sum_{j=k+1}^{R_1}P_j(\Lambda t)}\\&=\frac{\frac{1}{R_1+3-\Lambda t}\frac{R_1+3}{R_1+2}\left(R_1+2+\frac{\Lambda t}{R_1+3-\Lambda t}\right)P_{R_1+1}(\Lambda t)}{\sum_{j=k+1}^{R_1}P_j(\Lambda t)}\\&\leq\delta,\end{aligned}$$

which proves Ineq. (10) in the case $\sum_{j=0}^{k} P_j(\Lambda t) > 0.9$ and concludes the proof of Proposition 1. □

Proof of Proposition 2 From Ineq. 13, and Eqs. 14, and 12, the definition of the truncation parameter K_1 is seen to be equivalent to

$$K_1 = \min\left\{k \geq 0 : \; M_k - m_k \leq 2\frac{\sum_{j=0}^{k} v_j P_j(\Lambda t)}{1 - \sum_{j=0}^{k} P_j(\Lambda t)}\varepsilon + (m_k + M_k)(\varepsilon(1-\delta) - \delta)\right\}. \tag{46}$$

Then, since, by Ineq. 6, $M_k > 0$, $k \geq 0$, and, as assumed, $\delta < 1$ and $\varepsilon > \delta/(1-\delta)$, we have $(m_k + M_k)(\varepsilon(1-\delta) - \delta) > 0$, $k \geq 0$, which combined with, by Eq. 2,

$$\lim_{k\to\infty}(M_k - m_k) = \lim_{k\to\infty}\left(\max_i c_i^{(k)} - \min_i c_i^{(k)}\right) = 0\,, \tag{47}$$

implies that K_1 is finite.

It remains to prove the inequality

$$\left|\frac{\mathrm{ETRR}(t') - \widehat{\mathrm{ETRR}}(t', K_1))}{\widehat{\mathrm{ETRR}}(t', K_1)}\right| \leq \varepsilon\,.$$

To that end, we will start by bounding $|\widehat{\mathrm{ETRR}}(t', K_1)|$ from below. Using Eq. 11, that, by Ineq. 10, $[1 - \sum_{j=0}^{K_1} P_j(\Lambda t')]^{\mathrm{lb}} \geq (1-\delta)(1 - \sum_{j=0}^{K_1} P_j(\Lambda t'))$, that, as assumed, $\delta < 1$, and that, by Ineq. 6, $M_{K_1} > 0$, $k \geq 0$, we obtain

$$\begin{aligned}\left|\widehat{\mathrm{ETRR}}(t', K_1)\right| &= \left| r_{\max}\left(\sum_{j=0}^{K_1} v_j P_j(\Lambda t') + \frac{m_{K_1} + M_{K_1}}{2}\left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t')\right]^{\mathrm{lb}}\right)\right| \\ &= r_{\max}\left(\sum_{j=0}^{K_1} v_j P_j(\Lambda t') + \frac{m_{K_1} + M_{K_1}}{2}\left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t')\right]^{\mathrm{lb}}\right) \\ &\geq r_{\max}\left(\sum_{j=0}^{K_1} v_j P_j(\Lambda t') + \frac{m_{K_1} + M_{K_1}}{2}(1-\delta)\left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t')\right)\right) \\ &> 0. \end{aligned} \tag{48}$$

Now, by Lemma 1 with $l = 0$, $m = K_1$, $f(k) = v_k$, $\lambda = \Lambda t$, $n = K_1 + 1$, $g(k) = 1$, and $\lambda' = \Lambda t'$,

$$\frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t)}{1 - \sum_{j=0}^{K_1} P_j(\Lambda t)} = \frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t)}{\sum_{j=K_1+1}^{\infty} P_j(\Lambda t)} \leq \frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t')}{\sum_{j=K_1+1}^{\infty} P_j(\Lambda t')} = \frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t')}{1 - \sum_{j=0}^{K_1} P_j(\Lambda t')}. \tag{49}$$

Then, using Eqs. 1, 11 and Ineq. 3, all with $k = K_1$, using that, by Ineq. 10, $1 - \sum_{j=0}^{K_1} P_j(\Lambda t') - [1 - \sum_{j=0}^{K_1} P_j(\Lambda t')]^{\mathrm{lb}} \le \delta(1 - \sum_{j=0}^{K_1} P_j(\Lambda t'))$, and using Eq. 46 and Ineq. 49,

$$
\begin{aligned}
\frac{\left|\mathrm{ETRR}(t') - \widehat{\mathrm{ETRR}}(t', K_1)\right|}{r_{\max}} &= \left| \sum_{j=K_1+1}^{\infty} v^{(j)} P_j(\Lambda t') - \frac{m_{K_1} + M_{K_1}}{2} \left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right| \\
&= \left| \sum_{j=K_1+1}^{\infty} v^{(j)} P_j(\Lambda t') - \frac{m_{K_1} + M_{K_1}}{2} \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right) \right. \\
&\quad \left. + \frac{m_{K_1} + M_{K_1}}{2} \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') - \left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right) \right| \\
&= \left| \sum_{j=K_1+1}^{\infty} v^{(j)} P_j(\Lambda t') - \frac{m_{K_1} + M_{K_1}}{2} \sum_{j=K_1+1}^{\infty} P_j(\Lambda t') \right. \\
&\quad \left. + \frac{m_{K_1} + M_{K_1}}{2} \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') - \left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right) \right| \\
&= \left| \sum_{j=K_1+1}^{\infty} \left(v^{(j)} - \frac{m_{K_1} + M_{K_1}}{2} \right) P_j(\Lambda t') \right. \\
&\quad \left. + \frac{m_{K_1} + M_{K_1}}{2} \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') - \left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right) \right| \\
&\le \sum_{j=K_1+1}^{\infty} \left| v^{(j)} - \frac{m_{K_1} + M_{K_1}}{2} \right| P_j(\Lambda t') \\
&\quad + \frac{m_{K_1} + M_{K_1}}{2} \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') - \left[1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right) \\
&\le \frac{M_{K_1} - m_{K_1}}{2} \sum_{j=K_1+1}^{\infty} P_j(\Lambda t') + \frac{m_{K_1} + M_{K_1}}{2} \delta \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right) \\
&= \left(\frac{M_{K_1} - m_{K_1}}{2} + \delta \frac{m_{K_1} + M_{K_1}}{2} \right) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right) \\
&\le \left(\frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t)}{1 - \sum_{j=0}^{K_1} P_j(\Lambda t)} \varepsilon + \frac{m_{K_1} + M_{K_1}}{2} (\varepsilon(1 - \delta) - \delta) \right. \\
&\quad \left. + \delta \frac{m_{K_1} + M_{K_1}}{2} \right) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right)
\end{aligned}
$$

$$= \left(\frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t)}{1-\sum_{j=0}^{K_1} P_j(\Lambda t)} \varepsilon + \frac{m_{K_1}+M_{K_1}}{2} \varepsilon(1-\delta) \right) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right)$$

$$\leq \left(\frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t')}{1-\sum_{j=0}^{K_1} P_j(\Lambda t')} \varepsilon + \frac{m_{K_1}+M_{K_1}}{2} \varepsilon(1-\delta) \right) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right)$$

$$= \sum_{j=0}^{K_1} v_j P_j(\Lambda t')\, \varepsilon + \frac{m_{K_1}+M_{K_1}}{2} \varepsilon(1-\delta) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right) . \tag{50}$$

Finally, combining Ineqs. 48, 50,

$$\left| \frac{\mathrm{ETRR}(t') - \widehat{\mathrm{ETRR}}(t', K_1))}{\widehat{\mathrm{ETRR}}(t', K_1)} \right| \leq \frac{\sum_{j=0}^{K_1} v_j P_j(\Lambda t')\, \varepsilon + \frac{m_{K_1}+M_{K_1}}{2} \varepsilon(1-\delta) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right)}{\sum_{j=0}^{K_1} v_j P_j(\Lambda t') + \frac{m_{K_1}+M_{K_1}}{2} (1-\delta) \left(1 - \sum_{j=0}^{K_1} P_j(\Lambda t') \right)}$$
$$= \varepsilon .$$

This completes the proof of Proposition 2. □

Proof of Proposition 3 That the truncation parameter R_2 given by Eq. 28 is finite follows from

$$\lim_{r\to\infty} \frac{\frac{1}{r+3-\Lambda t} \frac{r+3}{r+2} \left(r+2+\frac{\Lambda t}{r+3-\Lambda t} \right) P_{r+1}(\Lambda t)}{\sum_{j=k+2}^{r} P_j(\Lambda t)} = 0$$

and the fact that, as assumed, $\delta > 0$. Similarly, that the truncation parameter R_3 given by Eq. 30 is finite follows from

$$\lim_{r\to\infty} \frac{\frac{1}{r+3-\Lambda t} \frac{r+3}{r+2} \left(r+1-k+\frac{\Lambda t}{r+3-\Lambda t} \right) P_{r+1}(\Lambda t)}{\sum_{j=k+1}^{r} \frac{j-k}{j+1} P_j(\Lambda t)} = 0$$

and the fact that $\delta > 0$.

In the case $\sum_{j=0}^{k+1} P_j(\Lambda t) \leq 0.9$, Ineq. (31) is trivially true. In the case $\sum_{j=0}^{k+1} P_j(\Lambda t) > 0.9$, we have, by Eq. 27,

$$0 \leq \frac{\sum_{j=k+2}^{\infty} P_j(\Lambda t) - \sum_{j=k+2}^{R_2} P_j(\Lambda t)}{\sum_{j=k+2}^{\infty} P_j(\Lambda t)} = \frac{1-\sum_{j=0}^{k+1} P_j(\Lambda t) - [1-\sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}}{1-\sum_{j=0}^{k+1} P_j(\Lambda t)} .$$

Besides, we necessarily have $k+1 \geq \Lambda t - \log 2$ because the median of a Poisson distribution with parameter Λt is nonsmaller than $\Lambda t - \log 2$ (Choi 1994). Therefore, by Eq. 28, we also have $R_2 \geq k+2 \geq \Lambda t - \log 2 + 1 > \Lambda t$, which implies $R_2 + 1 > \Lambda t - 2$, and can then bound $\sum_{j=R_2+1}^{\infty} P_j(\Lambda t)$ from above by invoking Lemma 2 with $r = R_2 + 1$, $n = 0$, $\lambda = \Lambda t$, $m = 1$, and $m' = 1$. The result is:

$$\sum_{j=R_2+1}^{\infty} P_j(\Lambda t) \leq \left(1 - \frac{\Lambda t}{R_2+3} \right)^{-1} \left(P_{R_2+1}(\Lambda t) + \frac{\Lambda t}{R_2+3} \frac{1}{\Lambda t} \left(1 - \frac{\Lambda t}{R_2+3} \right)^{-1} P_{R_2+2}(\Lambda t) \right) . \tag{51}$$

Then, using Eq. 27, Ineq. 51, $P_{R_2+2}(\Lambda t) = (\Lambda t/(R_2+2))P_{R_2+1}(\Lambda t)$, and Eq. 28,

$$
\begin{aligned}
&\frac{1-\sum_{j=0}^{k+1} P_j(\Lambda t) - [1-\sum_{j=0}^{k+1} P_j(\Lambda t)]^{\mathrm{lb}}}{1-\sum_{j=0}^{k+1} P_j(\Lambda t)} \\
&= \frac{\sum_{j=k+2}^{\infty} P_j(\Lambda t) - \sum_{j=k+2}^{R_2} P_j(\Lambda t)}{\sum_{j=k+2}^{\infty} P_j(\Lambda t)} \\
&= \frac{\sum_{j=R_2+1}^{\infty} P_j(\Lambda t)}{\sum_{j=k+2}^{\infty} P_j(\Lambda t)} \\
&< \frac{\sum_{j=R_2+1}^{\infty} P_j(\Lambda t)}{\sum_{j=k+2}^{R_2} P_j(\Lambda t)} \\
&\le \frac{\left(1-\frac{\Lambda t}{R_2+3}\right)^{-1}\left(P_{R_2+1}(\Lambda t) + \frac{\Lambda t}{R_2+3}\frac{1}{\Lambda t}\left(1-\frac{\Lambda t}{R_2+3}\right)^{-1} P_{R_2+2}(\Lambda t)\right)}{\sum_{j=k+2}^{R_2} P_j(\Lambda t)} \\
&= \frac{\frac{1}{R_2+3-\Lambda t}\frac{R_2+3}{R_2+2}\left(R_2+2+\frac{\Lambda t}{R_2+3-\Lambda t}\right)P_{R_2+1}(\Lambda t)}{\sum_{j=k+2}^{R_2} P_j(\Lambda t)} \\
&\le \delta\,,
\end{aligned}
$$

showing Ineq. (31) in the case $\sum_{j=0}^{k+1} P_j(\Lambda t) > 0.9$.

It remains to prove Ineq. (32). If $k+1 \le \Lambda t$, or $k+1 > \Lambda t$ and $1-(0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t) \le \sum_{j=0}^{k+1} P_j(\Lambda t) \le 0.9$, the inequality is trivially true. Otherwise, i.e., if $k+1 > \Lambda t$, and $1-(0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t) > \sum_{j=0}^{k+1}$ or $\sum_{j=0}^{k+1} P_j(\Lambda t) > 0.9$, we clearly have, by Eqs. 22, 29,

$$
\begin{aligned}
0 &\le \frac{\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t) - \sum_{j=k+1}^{R_3}\frac{j-k}{j+1}P_j(\Lambda t)}{\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)} \\
&= \frac{P_{k+1}(\Lambda t) + \frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1} P_j(\Lambda t)\right) - \left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)\right]^{\mathrm{lb}}}{P_{k+1}(\Lambda t) + \frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1} P_j(\Lambda t)\right)}\,.
\end{aligned}
$$

Besides, since, by Eq. 30, $R_3 \ge k+1 > \Lambda t$, implying $R_3+1 > \Lambda t - 2$, we can invoke Lemma 2 with $r = R_3+1$, $n = k+1$, $\lambda = \Lambda t$, $m = 1$, and $m' = 1$, obtaining

$$
\begin{aligned}
\sum_{j=R_3+1}^{\infty}\left(1-\frac{k+1}{j+1}\right)P_j(\Lambda t) \le \left(1-\frac{\Lambda t}{R_3+3}\right)^{-1}\Bigg(&\left(1-\frac{k+1}{R_3+2}\right)P_{R_3+1}(\Lambda t) \\
&+\frac{\Lambda t}{R_3+3}\frac{1}{\Lambda t}\left(1-\frac{\Lambda t}{R_3+3}\right)^{-1}P_{R_3+2}(\Lambda t)\Bigg)\,. \qquad (52)
\end{aligned}
$$

Then, using Eq. 22, 29, and Ineq. 52, $P_{R_3+2}(\Lambda t) = (\Lambda t/(R_3+2))P_{R_3+1}(\Lambda t)$, and Eq. 30,

$$
\begin{aligned}
&\frac{P_{k+1}(\Lambda t)+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)-\left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)\right]^{\text{lb}}}{P_{k+1}(\Lambda t)+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)}\\
&\quad=\frac{\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)-\sum_{j=k+1}^{R_3}\frac{j-k}{j+1}P_j(\Lambda t)}{\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t)}\\
&\quad=\frac{\sum_{j=R_3+1}^{\infty}\dfrac{j-k}{j+1}P_j(\Lambda t)}{\sum_{j=k+1}^{\infty}\dfrac{j-k}{j+1}P_j(\Lambda t)}\\
&\quad<\frac{\sum_{j=R_3+1}^{\infty}\dfrac{j-k}{j+1}P_j(\Lambda t)}{\sum_{j=k+1}^{R_3}\frac{j-k}{j+1}P_j(\Lambda t)}\\
&\quad=\frac{\sum_{j=R_3+1}^{\infty}\left(1-\dfrac{k+1}{j+1}\right)P_j(\Lambda t)}{\sum_{j=k+1}^{R_3}\dfrac{j-k}{j+1}P_j(\Lambda t)}\\
&\quad\le\frac{\left(1-\dfrac{\Lambda t}{R_3+3}\right)^{-1}\left(\left(1-\dfrac{k+1}{R_3+2}\right)P_{R_3+1}(\Lambda t)+\dfrac{\Lambda t}{R_3+3}\dfrac{1}{\Lambda t}\left(1-\dfrac{\Lambda t}{R_3+3}\right)^{-1}P_{R_3+2}(\Lambda t)\right)}{\sum_{j=k+1}^{R_3}\dfrac{j-k}{j+1}P_j(\Lambda t)}\\
&\quad\le\frac{\dfrac{1}{R_3+3-\Lambda t}\dfrac{R_3+3}{R_3+2}\left(R_3+1-k+\dfrac{\Lambda t}{R_3+3-\Lambda t}\right)P_{R_3+1}(\Lambda t)}{\sum_{j=k+1}^{R_3}\dfrac{j-k}{j+1}P_j(\Lambda t)}\\
&\quad\le\delta\,,
\end{aligned}
$$

which proves Ineq. (32) in the case $k+1>\Lambda t$, and $1-(0.9\Lambda t/(k+1-\Lambda t))P_{k+1}(\Lambda t)>\sum_{j=0}^{k+1}$ or $\sum_{j=0}^{k+1}P_j(\Lambda t)>0.9$ and concludes the proof of Proposition 3. □

Proof of Proposition 4 From Ineq. 35 and Eqs. 36, 12, the definition of the truncation point K_2 is readily seen to be equivalent to

$$
K_2=\min\left\{k\ge 0:\ M_k-m_k\le 2\frac{\sum_{j=0}^{k}w_jP_j(\Lambda t)}{P_{k+1}(\Lambda t)+\frac{\Lambda t-(k+1)}{\Lambda t}\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t)\right)}\varepsilon\right.
$$
$$
\left.+(m_k+M_k)\left(\varepsilon(1-\delta)-\delta\right)\right\}.
$$

Then, since, by Ineq. 6, $M_k>0$, $k\ge 0$, and, as assumed, $\delta<1$ and $\varepsilon>\delta/(1-\delta)$, we have $(m_k+M_k)(\varepsilon(1-\delta)-\delta)>0, k\ge 0$, which combined with by, Eq. 47, $\lim_{k\to\infty}(M_k-m_k)=0$, implies that K_2 is finite.

To show the inequality

$$
\left|\frac{\mathrm{EARR}(t')-\widehat{\mathrm{EARR}}(t',K_2)}{\widehat{\mathrm{EARR}}(t',K_2)}\right|\le\varepsilon\,,
$$

we will start by obtaining a suitable lower bound for $|\widehat{\mathrm{EARR}}(t', K_2)|/r_{\max}$. We have, by Ineq. 31,

$$\left[1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right]^{\mathrm{lb}} \geq (1-\delta)\left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right)$$

and, by Eq. 22 and Ineq. 32,

$$\left[\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\right]^{\mathrm{lb}} \geq (1-\delta)\left(\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\right).$$

Then, using Eq. 34 and the fact that, as assumed, $\delta < 1$,

$$\begin{aligned}
\left|\frac{\widehat{\mathrm{EARR}}(t', K_2)}{r_{\max}}\right| &= \left|\sum_{j=0}^{K_2} w_j P_j(\Lambda t') + w_{K_2}\frac{K_2+1}{\Lambda t'}\left[1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right]^{\mathrm{lb}}\right. \\
&\quad \left. + \frac{m_{K_2}+M_{K_2}}{2}\left[\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\right]^{\mathrm{lb}}\right| \\
&= \sum_{j=0}^{K_2} w_j P_j(\Lambda t') + w_{K_2}\frac{K_2+1}{\Lambda t'}\left[1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right]^{\mathrm{lb}} \\
&\quad + \frac{m_{K_2}+M_{K_2}}{2}\left[\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\right]^{\mathrm{lb}} \\
&\geq \sum_{j=0}^{K_2} w_j P_j(\Lambda t') + w_{K_2}\frac{K_2+1}{\Lambda t'}(1-\delta)\left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right) \\
&\quad + \frac{m_{K_2}+M_{K_2}}{2}(1-\delta)\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \\
&\triangleq f(t', K_2) \qquad (53) \\
&> 0\,.
\end{aligned}$$

Now, we will derive an upper bound for $|\mathrm{EARR}(t') - \widehat{\mathrm{EARR}}(t', K_2)|/r_{\max}$. To that end, we note that, given $k \geq 0$ and $j \geq k+1$, from Eq. 19 we obtain

$$\begin{aligned}
w_j &= \frac{1}{j+1}\left(\sum_{l=0}^{k} v_l + \sum_{l=k+1}^{j} v_l\right) \\
&= \frac{1}{j+1}\left(\sum_{l=0}^{k} v_l + ((j+1)-(k+1))\frac{m_k+M_k}{2} + \sum_{l=k+1}^{j}\left(v_l - \frac{m_k+M_k}{2}\right)\right) \\
&= \frac{k+1}{j+1} w_k + \frac{m_k+M_k}{2} - \frac{k+1}{j+1}\frac{m_k+M_k}{2} + \frac{1}{j+1}\sum_{l=k+1}^{j}\left(v_l - \frac{m_k+M_k}{2}\right). \qquad (54)
\end{aligned}$$

Then, using Eqs. 18, 34, 54, 22, and Ineq. 3,

$$
\begin{aligned}
\frac{|\mathrm{EARR}(t') - \widehat{\mathrm{EARR}}(t',k)|}{r_{\max}} &= \Bigg| \sum_{j=k+1}^{\infty} w_j P_j(\Lambda t') - \frac{k+1}{\Lambda t'} w_k \left[1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \\
&\quad - \frac{m_k + M_k}{2} \left[\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \Bigg| \\
&= \Bigg| \sum_{j=k+1}^{\infty} \frac{k+1}{j+1} w_k P_j(\Lambda t') + \frac{m_k + M_k}{2} \sum_{j=k+1}^{\infty} P_j(\Lambda t') \\
&\quad - \sum_{j=k+1}^{\infty} \frac{k+1}{j+1} \frac{m_k + M_k}{2} P_j(\Lambda t') \\
&\quad + \sum_{j=k+1}^{\infty} \frac{1}{j+1} \left(\sum_{l=k+1}^{j} \left(v_l - \frac{m_k + M_k}{2} \right) \right) P_j(\Lambda t') \\
&\quad - \frac{k+1}{\Lambda t'} w_k \left[1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right]^{\mathrm{lb}} - \frac{m_k + M_k}{2} \left[\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \Bigg| \\
&= \Bigg| \frac{k+1}{\Lambda t'} w_k \sum_{j=k+2}^{\infty} P_j(\Lambda t') + \frac{m_k + M_k}{2} \sum_{j=k+1}^{\infty} P_j(\Lambda t') \\
&\quad - \frac{k+1}{\Lambda t'} \frac{m_k + M_k}{2} \sum_{j=k+2}^{\infty} P_j(\Lambda t') \\
&\quad + \sum_{j=k+1}^{\infty} \frac{1}{j+1} \left(\sum_{l=k+1}^{j} \left(v_l - \frac{m_k + M_k}{2} \right) \right) P_j(\Lambda t') \\
&\quad - \frac{k+1}{\Lambda t'} w_k \left[1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right]^{\mathrm{lb}} - \frac{m_k + M_k}{2} \left[\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \Bigg| \\
&= \Bigg| \frac{k+1}{\Lambda t'} w_k \left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right) + \frac{m_k + M_k}{2} \left(1 - \sum_{j=0}^{k} P_j(\Lambda t') \right) \\
&\quad - \frac{k+1}{\Lambda t'} \frac{m_k + M_k}{2} \left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right) \\
&\quad + \sum_{j=k+1}^{\infty} \frac{1}{j+1} \left(\sum_{l=k+1}^{j} \left(v_l - \frac{m_k + M_k}{2} \right) \right) P_j(\Lambda t') \\
&\quad - \frac{k+1}{\Lambda t'} w_k \left[1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right]^{\mathrm{lb}} - \frac{m_k + M_k}{2} \left[\sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \Bigg| \\
&= \Bigg| \frac{k+1}{\Lambda t'} w_k \left(1 - \sum_{j=0}^{k+1} P_j(\Lambda t') \right) + \frac{m_k + M_k}{2} \sum_{j=k+1}^{\infty} \frac{j-k}{j+1} P_j(\Lambda t')
\end{aligned}
$$

$$+\sum_{j=k+1}^{\infty}\frac{1}{j+1}\left(\sum_{l=k+1}^{j}\left(v_l-\frac{m_k+M_k}{2}\right)\right)P_j(\Lambda t')$$

$$-\frac{k+1}{\Lambda t'}w_k\left[1-\sum_{j=0}^{k+1}P_j(\Lambda t')\right]^{\mathrm{lb}}-\frac{m_k+M_k}{2}\left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\Bigg|$$

$$=\Bigg|\frac{k+1}{\Lambda t'}w_k\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t')-\left[1-\sum_{j=0}^{k+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\right)$$

$$+\frac{m_k+M_k}{2}\left(\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')-\left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\right)$$

$$+\sum_{j=k+1}^{\infty}\frac{1}{j+1}\left(\sum_{l=k+1}^{j}\left(v_l-\frac{m_k+M_k}{2}\right)\right)P_j(\Lambda t')\Bigg|$$

$$\leq\frac{k+1}{\Lambda t'}w_k\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t')-\left[1-\sum_{j=0}^{k+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\right)$$

$$+\frac{m_k+M_k}{2}\left(\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')-\left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\right)$$

$$+\sum_{j=k+1}^{\infty}\frac{1}{j+1}\left(\sum_{l=k+1}^{j}\left|v_l-\frac{m_k+M_k}{2}\right|\right)P_j(\Lambda t')$$

$$\leq\frac{k+1}{\Lambda t'}w_k\left(1-\sum_{j=0}^{k+1}P_j(\Lambda t')-\left[1-\sum_{j=0}^{k+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\right)$$

$$+\frac{m_k+M_k}{2}\left(\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')-\left[\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\right)$$

$$+\frac{M_k-m_k}{2}\sum_{j=k+1}^{\infty}\frac{j-k}{j+1}P_j(\Lambda t')\,.$$

Now, replacing k by K_2 in the above inequality and using that, by Ineq. 31,

$$1-\sum_{j=0}^{K_2+1}P_j(\Lambda t')-\left[1-\sum_{j=0}^{K_2+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\leq\delta\left(1-\sum_{j=0}^{K_2+1}P_j(\Lambda t')\right),$$

that, by Ineq. 32 and Eq. 22,

$$\sum_{j=K_2+1}^{\infty}\frac{j-K_2}{j+1}P_j(\Lambda t')-\left[\sum_{j=K_2+1}^{\infty}\frac{j-K_2}{j+1}P_j(\Lambda t')\right]^{\mathrm{lb}}\leq\delta\sum_{j=K_2+1}^{\infty}\frac{j-K_2}{j+1}P_j(\Lambda t'),$$

Ineq. (3) with k replaced by K_2, that, from Ineq. 35, and Eqs. 36, 22, and 12,

$$M_{K_2} - m_{K_2} \leq 2 \frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t)}{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t)} \varepsilon + (m_k + M_k)\left(\varepsilon(1-\delta) - \delta\right) \ ,$$

and that, as assumed, $\delta < \varepsilon(1-\delta)$, gives

$$\begin{aligned}
\frac{|\mathrm{EARR}(t') - \widehat{\mathrm{EARR}}(t', K_2)|}{r_{\max}} &\leq \frac{K_2+1}{\Lambda t'} w_{K_2} \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t') - \left[1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right) \\
&\quad + \frac{m_{K_2} + M_{K_2}}{2} \left(\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') - \left[\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \right]^{\mathrm{lb}} \right) \\
&\quad \frac{M_{K_2} - m_{K_2}}{2} \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \\
&\leq \frac{K_2+1}{\Lambda t'} w_{K_2} \delta \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t') \right) + \frac{m_{K_2} + M_{K_2}}{2} \delta \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \\
&\quad + \frac{m_{K_2} - M_{K_2}}{2} \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \\
&\leq \frac{K_2+1}{\Lambda t'} w_{K_2} \delta \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t') \right) + \frac{m_{K_2} + M_{K_2}}{2} \delta \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \\
&\quad + \frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t)}{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t)} \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \varepsilon \\
&\quad + \frac{m_{K_2} + M_{K_2}}{2} \left(\varepsilon(1-\delta) - \delta\right) \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \\
&= \frac{K_2+1}{\Lambda t'} w_{K_2} \delta \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t') \right) \\
&\quad + \frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t)}{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t)} \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \varepsilon \\
&\quad + \frac{m_{K_2} + M_{K_2}}{2} (1-\delta) \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \varepsilon \\
&\leq \frac{K_2+1}{\Lambda t'} w_{K_2} \varepsilon (1-\delta) \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t') \right) \\
&\quad + \frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t)}{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t)} \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \varepsilon \\
&\quad + \frac{m_{K_2} + M_{K_2}}{2} (1-\delta) \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t') \varepsilon \, . \qquad (55)
\end{aligned}$$

Now, by Lemma 1 with $l = 0$, $m = K_2$, $f(k) = w_k$, $\lambda = \Lambda t$, $n = K_2 + 1$, $g(k) = (k - K_2)/(k + 1)$, and $\lambda' = \Lambda t'$,

$$\frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t)}{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t)} \le \frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t')}{\sum_{j=K_2+1}^{\infty} \dfrac{j-K_2}{j+1} P_j(\Lambda t')} . \tag{56}$$

Therefore, by Ineqs. 55 and 56, recalling Ineq. 53,

$$\begin{aligned}\frac{|\mathrm{EARR}(t') - \widehat{\mathrm{EARR}}(t', K_2)|}{r_{\max}} &\le \frac{K_2+1}{\Lambda t'} w_{K_2} \varepsilon (1-\delta) \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right) \\ &\quad + \frac{\sum_{j=0}^{K_2} w_j P_j(\Lambda t')}{\sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')} \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\, \varepsilon \\ &\quad + \frac{m_{K_2} + M_{K_2}}{2} (1-\delta) \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\, \varepsilon \\ &= \frac{K_2+1}{\Lambda t'} w_{K_2} \varepsilon (1-\delta) \left(1 - \sum_{j=0}^{K_2+1} P_j(\Lambda t')\right) + \sum_{j=0}^{K_2} w_j P_j(\Lambda t')\, \varepsilon \\ &\quad + \frac{m_{K_2} + M_{K_2}}{2} (1-\delta) \sum_{j=K_2+1}^{\infty} \frac{j-K_2}{j+1} P_j(\Lambda t')\, \varepsilon \\ &= \varepsilon f(t', K_2) . \end{aligned} \tag{57}$$

Combined with Eq. 53, Ineq. 57 gives

$$\left| \frac{\mathrm{EARR}(t') - \widehat{\mathrm{EARR}}(t', K_2)}{\widehat{\mathrm{EARR}}(t', K_2)} \right| \le \frac{\varepsilon f(t', K_2)}{f(t', K_2)} = \varepsilon .$$

This concludes the proof of Proposition 4. □

References

Abramowitz M, Stegun IA (1964) Handbook of mathematical functions: with formulas, graphs, and mathematical tables. Number 55 in Applied Mathematics Series. Courier Corporation. 10th printing (December 1972), with corrections

Bowerman PN, Nolty RG, Scheuer EM (1990) Calculation of the Poisson cumulative distribution function. IEEE Trans Reliab 39(2):158–161

Carrasco JA (2003a) Solving dependability/performability irreducible Markov models using regenerative randomization. IEEE Trans Reliab 52(3):319–329

Carrasco JA (2003b) Transient analysis of rewarded continuous time Markov models by regenerative randomization with Laplace transform inversion. Comput J 46(1):84–99

Carrasco JA (2004) Transient analysis of some rewarded Markov models using randomization with quasistationarity detection. IEEE Trans Comput 53(9):1106–1120

Carrasco JA (2015) Numerically stable methods for the computation of exit rates in Markov chains. Methodol Comput Appl Probab. To appear

Chen PM, Lee EK, Gibson GA, Katz RH, Patterson DA (1994) RAID: high-performance, reliable secondary storage. ACM Comput Surv 26(2):145–185

Choi KP (1994) On the median of gamma distributions and an equation of Ramanujan. Proc Am Math Soc 121:245–251

Fousse L, Hanrot G, Lefèvre V, Pélissier P, Zimmermann P (2007) MPFR: a multiple-precision binary floating-point library with correct rounding. ACM Trans Math Softw 33(2):13:1–13:15

Fox BL, Glynn PW (1988) Computing Poisson probabilities. Commun ACM 31(4):440–445
Glynn P (1987) Upper bounds on Poisson tail probabilities. Oper Res Lett 6(1):9–14
Grassmann WK (1977) Transient solutions in Markovian queueing systems. Comput Oper Res 4(1):47–53
Gross D, Miller DR (1984) The randomization technique as a modeling tool and solution procedure for transient Markov processes. Oper Res 32(2):343–361
IEEE754 (1985) IEEE standard for binary floating-point arithmetic. ANSI/IEEE Std 754-1985
IEEE754-2008 (2008) IEEE standard for floating-point arithmetic - redline. IEEE Std 754-2008 (Revision of IEEE Std 754-1985) - Redline
Kijima M (1997) Markov processes for stochastic modeling. Chapman & Hall, London
Knüsel L (1986) Computation of the chi-square and Poisson distribution. SIAM J Sci Stat Comput 7(3):1022–1036
Ogita T, Rump SM, Oishi S (2005) Accurate sum and dot product. SIAM J Sci Comput 26(6):1955–1988
Sericola B (1999) Availability analysis of repairable computer systems and stationarity detection. IEEE Trans Comput 48(11):1166–1172
Sidje RB, Burrage K, MacNamara S (2007) Inexact uniformization method for computing transient distributions of Markov chains. SIAM J Sci Comput 29(6):2562–2580
Stallman RM et al. (2012) Using the GNU compiler collection. For GCC version 4.7.2. Free Software Foundation
Suñé V, Carrasco JA (2005) Efficient implementations of the randomization method with control of the relative error. Comput Oper Res 32:1089–1114
van Moorsel APA, Sanders WH (1994) Adaptive uniformization. Stoch Model 10(3):619–647