Computing performability measures in Markov chains by means of matrix functions[☆]

G. Masetti[a,c], L. Robol[b,c,1,*]

[a] *Department of Computer Science, Largo B. Pontecorvo 3, Pisa, 56127, Italy.*
[b] *Department of Mathematics, Largo B. Pontecorvo 5, Pisa, 56127, Italy.*
[c] *Institute of Science and Technology "A. Faedo", Via G. Moruzzi, 1, 56124, Pisa, Italy.*

Abstract

We discuss the efficient computation of performance, reliability, and availability measures for Markov chains; these metrics — and the ones obtained by combining them, are often called performability measures.

We show that this computational problem can be recasted as the evaluation of a bilinear form induced by appropriate matrix functions, and thus solved by leveraging the fast methods available for this task.

We provide a comprehensive analysis of the theory required to translate the problem from the language of Markov chains to the one of matrix functions. The advantages of this new formulation are discussed, and it is shown that this setting allows to easily study the sensitivities of the measures with respect to the model parameters.

Numerical experiments confirm the effectiveness of our approach; the tests we have run show that we can outperform the solvers available in state of the art commercial packages on a representative set of large scale examples.

Keywords: Markov chains, Performance measures, Availability, Reliability, Matrix functions

1. Introduction

Performance and dependability models are ubiquitous [29] in design and assessment of physical, cyber or cyber-physical systems and a vast ecosystem of high level formalisms has been developed to enhance the expressive power

[☆] This research has been partially supported by the ISTI-CNR project "TAPAS: Tensor algorithm for performability analysis of large systems", by the INdAM/GNCS project 2018 "Tecniche innovative per problemi di algebra lineare", and by the Region of Tuscany (Project "MOSCARDO - ICT technologies for structural monitoring of age-old constructions based on wireless sensor networks and drones", 2016–2018, FAR FAS).

[*] Corresponding author

Email addresses: `giulio.masetti@isti.cnr.it` (G. Masetti), `leonardo.robol@unipi.it` (L. Robol)

[1] This author is a member of the INdAM Research group GNCS.

of Continuous Time Markov Chains (CTMCs). Examples include dialects of Stochastic Petri Nets (SPNs) such as Stochastic Reward Nets [29], Queuing networks [4], dialects of Performance Evaluation Process Algebra (PEPA) [18], and more. High level formalisms are to CTMC what high level programming languages are to machine code; in this setting, performance and dependability measures are usually defined following high level formalisms primitives. Once model and measures have been defined, automatic procedures synthesize, transparently to the modeler, a CTMC and a *reward structure* on it, producing a Markov Reward Process (MRP), and the measures of interest are derived as a function of the reward structure.

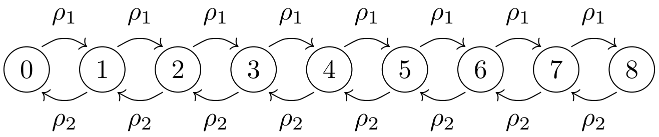
The main contribution of this paper is to show that the computation of these measures can be recasted in the framework of *matrix functions*, and therefore enable the use of fast Krylov-based methods for their computation. In particular, we provide a translation table that directly maps common performability measures to their matrix function formulation. Moreover, our approach is easily extendable to other kinds of measures. Matrix functions are a fundamental tool in numerical analysis, and arise in different areas of applied mathematics [17]. For instance, they are used in the evaluation of centrality measures for complex networks [12], in the computation of geometric matrix means that find applications in radar [5] and image processing techniques [14, 25], as well as in the study and efficient solution of system of ODEs [2, 19] and PDEs [31].

Many numerical problems in these settings can be rephrased as the evaluation of a bilinear form $g(v, w) = v^T f(A)w$, where $f(\cdot)$ is an assigned (matrix) function, A a matrix, and v and w vectors. Often, the interest is in the approximation of $g(v, w)$ for a specific choice of the arguments v, w, and so an explicit computation of $f(A)$ is both unnecessary and too expensive.

Therefore, one has to resort to more efficient methods, trying to exploit the structure of A when this is available. For instance, in [13] the authors describe an application to network analysis that involves a symmetric A (the adjacency matrix of an undirected graph), and they propose a Gauss quadrature scheme that provides guaranteed lower and upper bounds for the value of $g(v, w)$. Other approaches exploiting banded and rank structures in the matrices, often encountered in Markov chains, can be found in [6, 21]. These properties have already been exploited for the steady-state analysis of QBD processes [7, 8].

We prove that performability measures defined in CTMCs can be rephrased in this form as well. In particular they can be written as $g(v, w) = v^T f(Q)w$, where Q is the infinitesimal generator of the Markov chain (also called Markov chain transition matrix), and $f(\cdot), Q, v$, and w are chosen appropriately. CTMCs arise when modeling the behavior of resource sharing systems [4], software, hardware or cyber-physical systems [29], portfolio optimization [32], and the evaluation of performance [4], dependability [3, 28] and performability [23] measures that we are going to rewrite as bilinear forms $g(v, w)$. These models are obtained with different high-level formalism; however, all of them are eventually represented as CTMCs.

A simple example, which is analyzed in more detail in Section 6.1, can be constructed by a set of 9 states, numbered from 0 to 8, assuming that we can transition from state i to $i+1$ with an exponential distribution with parameter ρ_1, and from i to $i-1$ with rate ρ_2. Pictorially, this can be represented using the following reachability graph:


ρ_1 ρ_1 ρ_1 ρ_1 ρ_1 ρ_1 ρ_1 ρ_1
0 1 2 3 4 5 6 7 8
ρ_2 ρ_2 ρ_2 ρ_2 ρ_2 ρ_2 ρ_2 ρ_2

In this case, the infinitesimal generator matrix Q has the following structure:

$$Q = \begin{bmatrix} -\rho_2 & \rho_2 & & & \\ \rho_1 & -(\rho_1+\rho_2) & \rho_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \rho_1 & -(\rho_1+\rho_2) & \rho_2 \\ & & & \rho_1 & -\rho_1 \end{bmatrix},$$

where Q_{ij} is nonzero if and only if there is an edge from state i to j.

Rephrasing performability measures using the language of matrix functions opens also new direction of research regarding how structures that are visible at the level of these high level formalism are reflected into the CTMC infinitesimal generator, and then can be exploited to enhance measures' evaluation.

The paper is structured as follows. We start with a brief discussion of modeling and matrix function notations in Section 2.1. The main contribution is presented in Section 3, where we give a dictionary for the conversion of standard performance, dependability and performability measures in the parlance of matrix functions.

Using this new formulation, we present a new solution method in Section 4, and we demonstrate in Section 5 that this framework can be used to describe in an elegant and powerful form the *sensitivity* of the measures, which is directly connected to the Frechét derivative of $f(\cdot)$ at Q. We show that, when an efficient scheme for the evaluation of $g(v, w)$ is available, the sensitivity of the measures can be estimated at almost no additional cost.

Finally, we perform numerical tests on three relevant case studies in Section 6. The new method is shown to be efficient with respect to state-of-the-art solution techniques implemented in commercial level software tools. We also test our method to compute the sensitivity of measures as described in Section 5. The numerical experiments demonstrate that the computation requires less than twice the time needed to simply evaluate the measure.

To enhance readability, mathematical details are presented in the appendix. In particular, a brief discussion of the spectral properties of Q that are most relevant for our study is offered in appendix A.2; then, we give a summary of the currently available methods to compute performability measures in appendix A.3 and standard notions about matrix functions in appendix A.4. Details about the translation to matrix functions are presented in appendix Appendix B, where we provide a proof of the two main lemmas, Lemma 2.2 and Lemma 2.3, and we present additional remarks about their application.

2. Performability measures as matrix functions

2.1. Model and notation

We recall that Markov chains are stochastic processes with the *memory-less* property, that is the probability of jumping from state i to state j after some time t depends only on i and j, and not on the previous history. We consider CTMCs where the state space is finite so, without loss of generality, we assume it to be $[n] := \{1, \ldots, n\}$.

In addition, we assume that the probability of jumping from i to j is distributed with a given exponential rate λ_{ij}, so that we may define a matrix Q with entries

$$Q_{ij} = \begin{cases} \lambda_{ij} & \text{if } i \neq j \\ -(\lambda_{1i} + \ldots + \lambda_{n,i}) & \text{if } i = j \end{cases},$$

where we set $\lambda_{ii} = 0$ for any $i = 1, \ldots, n$. With this definition, given a certain initial probability distribution π_0^T, where the entry with index i corresponds to the probability of being in the state i, the probability at time t can be expressed as

$$\pi(t)^T = \pi_0^T e^{tQ}.$$

The stationary (or steady-state) distribution π, that is the limit of $\pi(t)$ for $t \to \infty$ is guaranteed[2] to exist if the process is *irreducible.* If the process has *absorbing states*, then the stationary distribution might not exists or not be unique; in this case, it is typically of interest to study the transient behavior of the process.

From the modeling perspective, the steady-state distribution describes the long-term behavior of a system. However, when assessing the performance and reliability of processes modeling real-world phenomena, it is essential to characterize the *transient* phase as well, that is the behavior between the initial configuration and the steady-state. Moreover, the transient state is relevant also when $X(t)$ is reducible, even though the steady-state distribution is not well-defined in this case.

In practice, if the system has a large number of states, computing $\pi(t)$ at some time t is not the desired measure; it is far more interesting to obtain concise information by "postprocessing" $\pi(t)$ in an appropriate way. Note that, in this framework, it is expected that $\pi(t)$ depends on the initial choice of π_0, so that is an important parameter that needs to be known.

Let us make an example to further clarify this concept. Consider a Markov chain $X(t)$, with infintesimal generator Q, modeling a publicly available service. Assume that we can partition the states in two sets $\mathbf{u}$ and $\mathbf{d}$. The first contains the states where the system is *online* ("up"), whereas the second the ones where the system is offline ("down"). Up to permuting the states, we can partition

[2] The uniqueness and existence of π is discussed in appendix A.1 in further detail.

the matrix Q according to this splitting:

$$Q = \begin{bmatrix} Q_{\mathbf{u}} & Q_{\mathbf{ud}} \\ Q_{\mathbf{du}} & Q_{\mathbf{d}} \end{bmatrix}.$$

In a certain interval of time $[0, t]$, we would like to know how long the system is expected to stay online. This can be measured by computing integral

$$U(t) = \int_0^t \mathbb{P}\{X(\tau) \in \mathbf{u}\} \; d\tau.$$

We note that $\mathbb{P}\{X(\tau) \in \mathbf{u}\} = \langle \pi(\tau), \mathbb{1}_{\mathbf{u}} \rangle$, where $\mathbb{1}_{\mathbf{u}}$ is the vector with ones in the states of $\mathbf{u}$, and 0 otherwise, and $\langle \cdot, \cdot \rangle$ denote the usual scalar product. This is what we call the *uptime* of the system in $[0, t]$. If the set $\mathbf{d}$ is the set of absorbing states, then this measure coincides with the *reliability* of the system.

This is an instance of a broader class of measures that belong to the field of *performance*, *availability* and *reliability* modeling. In the literature, the terms performance and availability refer to measures depending on the steady-state, whereas reliability concerns the transient phase of a reducible chain. Measures combining performance and availability (or reliability in the reducible case) are called *performability measures* [22].

This paper is concerned with the efficient computation of such measures, which will be obtained by rephrasing the problem as the evaluation of a bilinear form induced by a matrix function.

The main tool used to devise fast algorithms for the evaluation of these measures is recasting them as the computation of matrix functions. Informally, given any square matrix A and a function $f(z)$ which can be expanded as a power series, a matrix function $f(A)$ is defined as:

$$f(A) := \sum_{j \geq 0} c_j A^j, \qquad \text{where } f(z) = \sum_{j \geq 0} c_j z^j.$$

The most well-known example is the case $f(A) = e^A$, where $c_j = \frac{1}{j!}$. Another function of interest in this paper is $\varphi_1(z) := (e^z - 1)/z$. The definition holds in a more general setting, as discussed in appendix A.4.

Let us fix some notation. We denote by r a fixed weight vector of n elements, so that $r_{X(t)}$ is a process whose value at time t corresponds to entry of index $X(t)$ in r. In most cases, r will be the *reward vector*, containing a "prize" assigned for being a certain state. We shall give two definitions of reward measures, to simplify the discussion of their computation later on.

Definition 2.1. Let r be a reward vector, and $X(t)$ a Markov chain with infinitesimal generator Q. Then, the number $\mathbb{E}[r_{X(t)}]$ is called *instantaneous reward measure at time* t, and is denoted by $M_{\text{inst}}(t)$. Similarly, the number

$$M(t) = \int_0^t \mathbb{E}\left[r_{X(\tau)}\right] \; d\tau = \int_0^t M_{\text{inst}}(\tau) \; d\tau$$

is called *cumulative reward measure at time* t.

Note that both definitions depend on the choice of the reward vector r. This dependency is not explicit in our notation to make it more readable — in most of the following examples the current choice of r will be clear from the context. Occasionally, we will make this dependency explicit by saying that a measure is associated with a reward vector r.

Intuitively, the instantaneous reward measures the probability of being in a certain set of states (the non-zero entries of r), weighted according to the values of the components of r. The cumulative version is averaged over the time interval $[0, t]$. For instance, for the uptime we had $r := \mathbb{1}_{\mathbf{u}}$.

In the remaining part of this section, we show that reward measures can be expressed as $\pi_0^T f(Q)r$, for a certain reward vector r and an appropriate function $f(z)$.

The next result is the first example of this construction, and concerns instantaneous reward measures.

Lemma 2.2. *Let $M_{\text{inst}}(t)$ be an instantaneous reward measure associated with a vector r and a Markov process generated by Q and with initial state π_0. Then, $M_{\text{inst}}(t) = \pi_0^T f(Q)r$, with $f(z) := e^{tz}$.*

Proof. The equality follows immediately by the relation $\pi(t)^T = \pi_0^T e^{tQ}$. □

A similar statement holds for the cumulative reward measure.

Lemma 2.3. *Let M be an cumulative reward measure, as in Definition 2.1. Then, $M(t) = \pi_0^T f(Q)r$, with*

$$f(z) := t\varphi_1(tz) = \begin{cases} \frac{e^{tz}-1}{z} & z \neq 0 \\ t & z = 0 \end{cases}$$

Proof. The proof of this Lemma is given in Appendix B. □

3. Translation of performability measures into matrix functions

The purpose of this section is to construct a ready-to-use dictionary for researchers involved in modeling that can be used to translate several known measures to the matrix function formulation with little effort. This is achieved by applying some theoretical results which, to ease the reading, are discussed in Appendix B. Following these results, one can derive analogous formulations for additional performability measures.

For many measures, it is important to identify a set of states that correspond to the *online* or *up* state: when the system is in one of those states then it is functioning correctly. To keep a uniform notation, we refer to this set as $\mathbf{u} \subseteq \{1, \ldots, n\}$. Its complement, the *offline* or *down* states, will be denoted by $\mathbf{d}$. In the case of reducible Markov chains, the set of "down" states typically coincides with the absorbing ones. This hypothesis is necessary for some measures (such as the mean time to failure) in order to make them well-defined.

Clearly, the actual meaning of being "up" or "down" may change dramatically from one setting to another; but the computations involved are essentially unchanged — and therefore we prefer to keep this nomenclature to present a unified treatment. A summary of all the different reformulations in this section, with references to the location where the details are discussed, is given in Table 1.

3.1. Instantaneous reliability

We consider the case of a reducible chain, with a set of absorbing states (the "down" states). We are concerned with determining the probability of being in an "up" state at any time t. This measure is called *instantaneous reliability.*

This measure can be computed considering the probability of being in a state included in the set $\mathbf{u}$, i.e.,

$$R(t) = \mathbb{P}\{X(t) \in \mathbf{u}\} = \sum_{i \in \mathbf{u}} \pi_i(t) = \pi(t)^T r,$$

where $r = \mathbb{1}_{\mathbf{u}}$ is the vector with components 1 on the indices in $\mathbf{u}$, and zero otherwise. Therefore, this availability measure is rephrased in matrix functions terms as $R(t) = \pi_0^T e^{tQ} r$.

In the same way, we may define the measure $F(t) = 1 - R(t)$, which is the probability of being in "down" state at the time t. Notice that this can also be expressed in matrix function form by $F(t) = \pi_0^T e^{tQ}(\mathbb{1} - r)$.

3.2. Instantaneous availability

In irreducible Markov chains, the *instantaneous availability* is the analogue of the reliability described in the previous section, that is, we measure the probability of the system being "up" at any time t. We note that, mathematically, the definition of reliability and availability coincide but the former term is considered when dealing with reducible Markov chains, whereas the latter is employed for irreducible ones. In particular, the availability can be expressed in matrix function form as follows: $A(t) = \pi_0^T e^{tQ} \mathbb{1}_{\mathbf{u}}$.

When the states in $\mathbf{u}$ correspond to the working state of at least k components out of n, this measure is often called the *k-out-of-n availability* of the system.

3.3. Mean time to failure

We consider the expected time of failure for a model. This measure is relevant for devices which fail, and cannot be repaired. In particular, it is possible to exit from states in $\mathbf{u}$, but one can never go back again: the Markov chain is reducible and $\mathbf{d}$ is a set of absorbing states. The average time needed to exit $\mathbf{u}$ can be expressed as the average time that one spends inside $\mathbf{u}$. In probabilistic terms,

$$\text{MTTF} = \int_0^\infty \mathbb{E}[(\mathbb{1}_{\mathbf{u}})_{X(\tau)}]\, d\tau,$$

Measure	Function	Reward vector	Matrix	Reference
Inst. reliability	e^{tz}	$\mathbb{1}_{\mathbf{u}}$	Q	Section 3.1
Inst. availability	e^{tz}	$\mathbb{1}_{\mathbf{u}}$	Q	Section 3.2
MTTF	$-\frac{1}{z},\ t\varphi_1(tz)$	$\mathbb{1}_{\mathbf{u}}$	$Q_{\mathbf{u}},\ Q$	Section 3.3
Exp. # failures	$t\varphi_1(tz)$	$\mathbb{1}$	$Q_{\mathbf{ud}}$	Section 3.4
Uptime	$t\varphi_1(tz)$	$\mathbb{1}_{\mathbf{u}}$	Q	Section 3.5
Average clients	$e^{tz}, \delta(z)$	$[0, 1, \ldots, n-1]^T$	Q	Section 3.6

Table 1: Summary of the equivalence between performability measures and matrix functions, with the corresponding reward vector. The details on the interpretation of the set **u** and on the reformulation are given in the linked sections.

where $(\mathbb{1}_{\mathbf{u}})_{X(\tau)}$ denotes the component of index $X(\tau)$ in the vector $\mathbb{1}_{\mathbf{u}}$. For this measure to be finite, it is necessary that all the states inside of **u** have zero probability in the steady-state. This can be rephrased using $\pi(t)$ as follows:

$$\mathrm{MTTF} = \sum_{i\in\mathbf{u}} \int_0^\infty \pi_i(\tau)\ d\tau.$$

Here, one could be tempted to apply Lemma 2.3 directly, but this is not feasible. In fact, taking the limit of t to ∞ for $f(z)$ gives $f(z) = z^{-1}$, which has a pole at 0, and Q is always singular. However, one can notice that, for $i \in \mathbf{u}$, we have $\pi_i(t) = (\pi_{0,\mathbf{u}}^T e^{tQ_{\mathbf{u}}})_i$ where $\pi_{0,\mathbf{u}}$ is the vector of initial conditions restricted to the indices in **u**. Therefore, we can apply Lemma 2.3 and take the limit of $t \to \infty$ to obtain:

$$\mathrm{MTTF} = \pi_{0,\mathbf{u}}^T f(Q_{\mathbf{u}})\mathbb{1}, \qquad f(z) = -\frac{1}{z},$$

which can be written simply as $MTTF = -\pi_{0,\mathbf{u}}^T Q_{\mathbf{u}}^{-1}\mathbb{1}$. The matrix $Q_{\mathbf{u}}$ is invertible[3], its inverse is nonnegative [24], and therefore $MTTF > 0$.

The same measure is often restricted to the interval $[0, t]$. In this case, we may write

$$\mathrm{MTTF}(t) = \int_0^t \mathbb{E}[(\mathbb{1}_{\mathbf{u}})_{X(\tau)}]\ d\tau.$$

We note that this formulation is well-defined even when t goes to infinity. In fact, a direct application of Lemma 2.3 yields

$$\mathrm{MTTF}(t) = \pi_0^T f(Q)\mathbb{1}, \qquad f(z) = t\varphi_1(tz) = \frac{e^{tz}-1}{z},$$

and this function does not have a pole in 0. Nevertheless, also in this case it holds true that $\pi_0^T f(Q)\mathbb{1} = \pi_{0,\mathbf{u}}^T f(Q_{\mathbf{u}})\mathbb{1}$, and this gives a reduction in the size of the matrix whose exponential needs to be computed, so this reformulation may be convenient in practice.

[3] See Lemma A.4

3.4. Expected number of failures

We consider a system partitioned as usual in up and down states (denoted by $\mathbf{u}$ and $\mathbf{d}$). We are interested in computing the expected number of transitions between a state in $\mathbf{u}$ to a state in $\mathbf{d}$ (the number of failures).

If we consider two states i and j, then the expected number of transitions $N_{ij}(t)$ from i to j in a certain time interval $[0,t]$ can be expressed as

$$\mathbb{E}[N_{ij}(t)] = Q_{ij} \cdot \int_0^t \pi_i^T(\tau)\ d\tau.$$

When considering two sets of states, $\mathbf{u}$ and $\mathbf{d}$, this can be generalized to the expected number of transitions from $\mathbf{u}$ to $\mathbf{d}$ by

$$\int_0^t \pi(\tau) \begin{bmatrix} 0 & Q_{\mathbf{ud}} \\ 0 & 0 \end{bmatrix} \mathbb{1}\ d\tau = \int_0^t \pi_{\mathbf{u}}(\tau) Q_{\mathbf{ud}} \mathbb{1}\ d\tau = \pi_{0,\mathbf{u}}^T f(Q_{\mathbf{ud}})\mathbb{1},$$

where $f(z) = \frac{e^{tz}-1}{z}$, $\pi_{\mathbf{u}}(t)$ and $\pi_{0,\mathbf{u}}$ are the probability distributions restricted to the states in $\mathbf{u}$.

3.5. Uptime

The *uptime* measure determines the expected availability of a system in a time interval $[0,t]$, for an irreducible Markov chain. To this end, we need to partition the states in online and offline, and to compute the integral

$$U(t) = \int_0^t \mathbb{E}\left[r_{X(\tau)}\right]\ d\tau, \qquad r = \mathbb{1}_{\mathbf{u}}.$$

We note that this is the integral analogous of the instantaneous availability defined for irreducible systems in Section 3.2. In fact, a straightforward computation shows that

$$U(t) = \int_0^t A(\tau)\ d\tau = \pi_0^T f(Q)\mathbb{1}_{\mathbf{u}}, \qquad f(z) = t\varphi_1(tz) = \frac{e^{tz}-1}{z},$$

as predicted by Lemma 2.3. We notice that this measure is not well-defined if we let t go to infinity, since for every irreducible Markov chain $\lim_{t\to\infty} A(t) = \sum_{i\in\mathbf{u}} \pi_i > 0$, and therefore the limit of $U(t)$ needs to be infinite, because the integrand is not infinitesimal.

3.6. Average number of clients

We now discuss a measure which is specifically tailored to a model but, with the proper adjustments, can be made fit a broad number of settings. Assume we have a Markov chain $X(t)$ that models a queue (which might be at some desk serving clients, a server running some software, or similar use cases). The state of $X(t)$ is the number of clients waiting in the queue, and we assume a maximum number $n-1$ of slots. At any time, the process can finish to serve a client with a rate ρ_1, or get a new client in the queue with rate ρ_2.

We are interested in the expected number of clients in the queue at time $t > 0$, or at the steady state (that corresponds to $t \to \infty$). In the two cases, this measure can be expressed as

$$\mathbb{E}[X(t)] = \pi(t)^T v, \qquad \mathbb{E}[X(\infty)] = \pi^T v, \qquad v = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ n-1 \end{bmatrix},$$

where as usual we denote by $X(\infty)$ the limit of $X(t)$ to the steady-state. It is clear that, since $\pi(t) = \pi_0^T e^{tQ}$, we can express $\mathbb{E}[X(t)]$ as the bilinear form $\pi_0^T e^{tQ} v$. It is interesting that one can express the steady state probability in matrix function form as well. In fact, if we define $\delta(z)$ as the function equal to 1 at 0 and 0 elsewhere, we can express[4] $\mathbb{E}[X(\infty)]$ as $\pi_0^T \delta(Q) v$.

4. Efficient computation of the measures

In view of the analysis of Section 3, we are now aware that several measures associated with a Markov process $X(t)$ are in fact computable by evaluating $w^T f(Q) v$, for appropriate choices of w, v and of the matrix function $f(Q)$. A straightforward application of standard dense linear algebra methods to compute $f(Q)$ usually has complexity $\mathcal{O}(n^3)$, where n is the size of the matrix Q, which in this case is the number of states of the underlying Markov chain.

It is often recognized in the literature that the matrix Q generating the Markov chain is structured, and allows for a fast matrix vector product $v \mapsto Qv$. Typically, we can expect this operation to cost $\mathcal{O}(n)$ flops, where n is the number of states in the Markov chain. In this section we propose to leverage well established Krylov approximation methods for the computation of $w^T f(Q)$, which in turn yields an algorithm for evaluating $w^T f(Q) v$ in linear time and memory. Similar ideas and techniques can be found in *exponential integrators*, see [16].

Here we recall only the essential details needed to carry out the scheme, and we refer to [15] and the references therein for further details. We now focus on the computation of $f(Q)v$, ignoring w. Once this is known, $w^T f(Q) v$ can be obtained in $\mathcal{O}(n)$ flops through a scalar product.

4.1. Krylov subspace approximation

The key ingredient to the fast approximation of $f(Q)v$ is the so-called Arnoldi process, the non-symmetric extension of the Lanczos scheme. From now on, we assume without loss of generality that $\|v\|_2 = 1$. Consider the Krylov subspace of order m generated by Q and v as

$$\mathcal{K}_m(Q, v) := \operatorname{span}\{v, Qv, Q^2 v, \ldots, Q^{m-1} v\}.$$

[4] This is proven in Lemma B.2.

Assuming no breakdown happens, the Arnoldi scheme provides an orthogonal basis V_m for this space that satisfies the relation:

$$QV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T,$$

where H_m is an $m \times m$ upper Hessenberg matrix, $h_{m+1,m}$ a scalar, v_{m+1} a vector, and e_m the m-th column of the identity matrix. This relation is often used in the description of the classical Arnoldi method, and can be employed to iteratively and efficiently construct the basis V_m. We refer the reader to [11] for further details. This can be used to retrieve an approximation of $f(Q)v$ by computing $f_m = V_m f(H_m) V_m^T v = V_m f(H_m) e_1$. This approximation has several neat features, among which we find the *exactness* properties: the approximation f_m is exact if $f(z)$ is a polynomial of degree at most $m-1$. We would like to characterize how accurate is this approximation for generic function. To this aim, we introduce the following concept.

Definition 4.1. Given a matrix Q, the subset of the complex plane defined as

$$\mathcal{W}(Q) := \{x^T Q x \ : \ \|x\|_2 = 1\}$$

is called the *field of values* of Q.

The above set is easily seen to be convex, and always contain the eigenvalues of Q. Whenever Q is a normal matrix (for instance, when Q is symmetric), then the set $\mathcal{W}(Q)$ is the convex hull of the eigenvalues. For non-normal matrices, this set is typically slightly larger, but the following relation holds:

$$\sigma(Q) \subseteq \mathcal{W}(Q) \subseteq \{z \in \mathbb{C} \ : \ |z| \leq \|Q\|_2\} =: B(0, \|Q\|_2),$$

where $\sigma(Q)$ is the spectrum, i.e., the set of eigenvalues, of Q. so in particular the set is not unbounded and, in the Markov chain setting, it is typically to estimate $\|Q\|_2$ to obtain a rough approximation of its radius.

Lemma 4.2. *Let $f(z)$ be a function defined on the field of values of Q, and $p(z)$ a polynomial approximant of $f(z)$ such that $|f(z) - p(z)| \leq \epsilon$ on $\mathcal{W}(Q)$. Then,*

$$\|p(Q) - f(Q)\|_2 \leq (1 + \sqrt{2}) \cdot \epsilon.$$

Proof. This inequality follows immediately from the well-known Crouzeix inequality (sometimes called Crouzeix conjecture, since the bound is conjectured to hold with 2 in place of $1 + \sqrt{2}$). See, for instance, [9]. □

A straightforward implication of the above result is that if $p_m(z)$ is a degree $m-1$ approximant to $f(z)$ and $|p_m - f| \leq \epsilon_m$ on a domain containing $\mathcal{W}(Q)$, then $\|f_m - f(Q)v\|_2$ can be bounded by writing $f(z) = p_m(z) + r_m(z)$ and using the exactness property:

$$\begin{aligned}\|f_m - f(Q)v\|_2 &\leq \|V_m p_m(H_m)e_1 - V_m r_m(H_m)e_1 - p_m(Q)v + r_m(Q)v\|_2 \\ &\leq \|V_m r_m(H_m)e_1\|_2 + \|r_m(Q)v\|_2 \leq 2(1+\sqrt{2}) \cdot \epsilon_m.\end{aligned}$$

We know, for instance by Weierstrass' theorem, that polynomials approximate uniformly continuous functions on a compact set, so this alone guarantees convergence of the scheme. However, it tells us very little about the convergence speed. It turns out that, for many functions of practical use that have a high level of smoothness (such as $f(z) = e^z$), the convergence is fast.

When the dimension of the space $\mathcal{K}_m(Q, v)$ increases, the orthogonalization inside the Arnoldi scheme can become the dominant cost in the method. For this reason, it is advisable to employ restarting techniques. These aim at stopping the iteration after m becomes sufficiently large, and consider a partial approximation of the function $f_m^{(1)}$. Then, the residual $f(Q)v - f_m^{(1)}$ is approximated by restarting the Arnoldi scheme from scratch. The efficient and robust implementation of this scheme is non-trivial; we use the approach developed in [15], to which we refer the reader for further details on the topic.

4.2. Approximation of exponential and $\varphi_1(z)$

After running the Arnoldi scheme, we are left with a simpler problem: we need to compute $f(H_m)e_1$, where H_m is a small $m \times m$ matrix. In our case, we are interested in the functions

$$e^{tz} \qquad \text{and} \qquad t\varphi_1(tz) = \frac{e^{tz} - 1}{z}.$$

The literature on the efficient approximation of the matrix exponential is vast; the most common approach for e^{tz} is to use a Padé approximation scheme coupled with a scaling and squaring technique, which is the default method implemented by MATLAB through the function `expm`; see the discussion in [17] for the optimal choice of parameters for the scaling phase and the approximation rule. Then, one can consider the rational approximant to e^z obtained using the Padé scheme with order (d, d), let us call it $r(z)$, and so we have

$$e^{H_m} \approx \left(r\left(\frac{1}{2^h} H_m \right) \right)^{2^h}.$$

The latter matrix power can be efficiently computed by h steps of repeated squaring, and the evaluation of the rational function requires $\mathcal{O}(d)$ matrix multiplications and one inversion. The order d has to be chosen depending on the level of squaring (i.e., on the value of h), and is an integer between 6 and 13 (parameter tuning for optimal performance and accuracy can be a tricky task, so we suggest to either refer to [17] or to rely on the MATLAB implementation of `expm`).

Concerning the computation of $\varphi_1(z)$, we use a trick widely used in exponential integrators. In particular, we propose to recast the problem as the computation of a matrix exponential, by exploiting the following known result from the framework of exponential integrators, whose proof can be found in [2, Theorem 2.1].

Theorem 4.3. *Let A be any $n \times n$ square matrix, and $v \in \mathbb{C}^n$. Then, the following relation holds:*

$$\tilde{A} := \begin{bmatrix} A & v \\ 0_{1\times n} & 0 \end{bmatrix}, \qquad \begin{bmatrix} I_n & 0_{n\times 1} \end{bmatrix} e^{\tilde{A}} \begin{bmatrix} v \\ 0 \end{bmatrix} = \varphi_1(A)\ v.$$

The above result tells us that the action of $\varphi_1(A)$ on a vector v can be obtained by computing the action of a slightly larger matrix $\tilde{A}$ on the vector v padded with a final zero. Even when the norm of A is large, the techniques in [15] allow to accurately control the approximation error.

4.3. Incorporating restarting

In practice, the dimension of the Krylov space needed to achieve a satisfactory accuracy might be high, and therefore a more refined technique is needed to achieve a low computational cost. One of the most efficient techniques is to incorporate a *restarting scheme*: we stop the method after the dimension of the space reaches a certain maximum allowed dimension, obtaining an approximation $f_1(Q)v$ of low-quality. Then, we restart the method to approximate $(f - f_1)(Q)v$, i.e., the residual. The procedure is then repeated until convergence.

An efficient implementation of such scheme is far from being trivial, and we rely on the restarting scheme proposed in [15], to which we refer for further details. Our implementation relies on the `funm_quad` package that is provided accompanying the paper [15].

5. Sensitivity analysis

Performance and dependability models are often parametric, in the sense that some transition rates Q_{ij} can be functions of some parameter λ. As described in [26], the sensitivity analysis is the study of how the measures of interest vary at changing p.

5.1. A motivation for sensitivity analysis

During the design phase, a key requirement is to isolate the set of parameters that most influence the behavior of the system. This can guide optimization to the design. In particular, this allows to investigate the return of an investment aimed at changing some components, in terms of enhanced reliability and/or availability. When the budget for developing a new product is limited, this is of paramount importance.

Moreover, real world parameters come from actual (physical) measurements and therefore might be affected by measure errors of different orders of magnitude, in particular for cyber-physical systems. Sensitivity analysis can guide the effort in collecting the most relevant parameters with high precision and the other parameters with acceptable precision.

Another setting where sensitivity analysis plays a relevant role is the modeling of complex systems, where certain aspects of the system behavior are often abstracted away because the time scale at which they appear is considered too fine (avoiding stiffness) or in order to maintain a reasonable level of complexity within the model itself (state space explosion avoidance). In particular, a hierarchical modeling strategy [28] may be adopted: specific system components are modeled in isolation, measures are defined on them and the numerical value obtained evaluating the measures are used as parameters for the overall system model. The hierarchical strategy can be employed whenever the system logical structure presents a (partial) order among components and can involve several layers. Establishing to which extent each layer is sensitive to those parameters that come from an underlying layer enhances and guides modeling choices.

5.2. Sensibility analysis and Frechét derivatives

We are interested in bounding the first order expansion of a measure $g(v, w)$ when the infinitesimal generator changes along a certain direction. Let $g_p(v, w)$ be a performability measure of a system with matrix Q depending on a parameter p in a smooth way. We want to determine a real positive number M such that:

$$|g_p(v,w) - g_{p_0}(v,w)| \leq M \cdot |p - p_0| + \mathcal{O}(|p-p_0|^2).$$

This characterizes the amplification of the changes in the system behavior when the parameter p changes. If Q depends smoothly on p then we can expand it around p_0:

$$Q(p) = Q(p_0) + (p - p_0) \cdot \frac{\partial}{\partial p} Q(p_0) + R(p), \qquad \|R(p)\| \leq \mathcal{O}(|p-p_0|^2).$$

From now on, by a slight abuse of notation, we will write $O(|p-p_0|^2)$ in place of $R(p)$, meaning that the bound is correct up to the second order terms in norm. A straightforward computation yields the following result.

Lemma 5.1. *Let $Q(p)$ a matrix with a C^1 dependency on p around a point p_0, and let $g_p(v, w) = v^T f(Q(p))w$. Then, we have*

$$|g_p(v,w) - g_{p_0}(v,w)| \leq \left| v^T D_f(Q(p_0)) \left[\frac{\partial Q(p_0)}{\partial p} \right] w \right| + \mathcal{O}(|p-p_0|^2),$$

where $D_f(\cdot)$ is the Frechét derivative of the matrix function $f(\cdot)$.

Even more interestingly, there exists a simple strategy (presented, for example, in [17]) to compute the Frechét derivative along a certain direction by making use of block matrices. Specializing it to our case yields the following corollary.

Corollary 5.2. *Let $Q(t)$ a matrix with a C^1 dependency on p around a point p_0, and let $g_p(v, w) = v^T f(Q(p))w$. Then, we have*

$$|g_p(v,w) - g_{p_0}(v,w)| \leq \begin{bmatrix} v^T & 0^T \end{bmatrix} f\left(\begin{bmatrix} Q(p_0) & \frac{\partial Q(p_0)}{\partial p} \\ & Q(p_0) \end{bmatrix} \right) \begin{bmatrix} 0 \\ w \end{bmatrix} + \mathcal{O}(|p-p_0|^2),$$

where the vectors are partitioned accordingly to the 2×2 block matrix.

Proof. The result is an immediate consequence of [17, Theorem 4.12]. □

In view of the above result, if we are given an efficient method to evaluate $v^T f(Q)w$, the sensitivity with respect to a certain perturbation of parameters can be computed by extending the method to work on a matrix of double the dimension.

For dense linear algebra methods, which have a cubic complexity, this amounts to 8 times the cost of just computing $g(v, w)$; for methods based on quadrature or Krylov subspaces, which have a linear complexity in the dimension, this means twice the cost of an evaluation.

In both cases, the asymptotic cost for the computation does not increase.

6. Numerical tests

In this section we report some practical example of the computation of availability and performance measures relying on matrix functions.

The results can be replicated using the MATLAB code that we have published at `https://github.com/numpi/markov-measures`, by running the scripts `Example1.m`, ..., `Example4.m`. The numbering of the examples coincides with the one of the following subsections. The parameters controlling the truncation both in the commercial solver employed and in `funm_quad` are set to 10^{-8}. For the `funm_quad` package we have used a restart every 15 iterations and a maximum number of restarts equal to 10, which has never been reached in the experiments.

For instantaneous measures, our tests rely directly on the integral representation of the matrix exponential in `funm_quad`. This method is denoted by `quad_exp` in tables and figures. For cumulative measures, involving the evaluation of $\varphi_1(z)$, the methods based on rephrasing the problem as the action of a matrix exponential is identified by `exp_phi`.

The tests have been performed with MATLAB r2017b running on Ubuntu 17.10 on a computer with an Intel i7-4710MQ CPU running at 2.50 GHz, and with 16 GB of RAM clocked at 1333 MHz.

6.1. Average queue length

We consider a simple Markov chain that models a queue for some service. The process $X(t)$ has as possible states the integers $\{0, \dots, n-1\}$, which represent the number of clients in the queue. A pictorial representation of the states for $n = 9$ is given in Figure 1.

At any state, the rate of probability of jumping "left" (i.e., to serve one client) is equal to ρ_1, whereas the rate of probability at which a new client arrives is equal to ρ_2. The corresponding matrix Q for the Markov chain is as

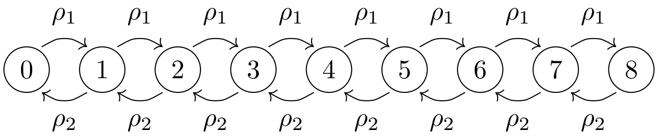


Figure 1: Pictorial representation of the Markov chain modeling a queue for a service, with $n = 9$ states. The rates of the probabilities of jumping between the states are reported on the edges.

follows:

$$Q = \begin{bmatrix} -\rho_2 & \rho_2 & & & \\ \rho_1 & -(\rho_1+\rho_2) & \rho_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \rho_1 & -(\rho_1+\rho_2) & \rho_2 \\ & & & \rho_1 & -\rho_1 \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

According to Section 3.6, this measure can be expressed in the form

$$M_{\text{inst}}(t) = \pi_0^T e^{tQ} r, \qquad r = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ n-1 \end{bmatrix}.$$

The reward vector r gives to each state a weight proportional to the number of clients waiting in the queue. We assume the initial state π_0 to be the vector e_1, corresponding to starting with an empty queue. The measure gives the average expected number of waiting clients at time t.

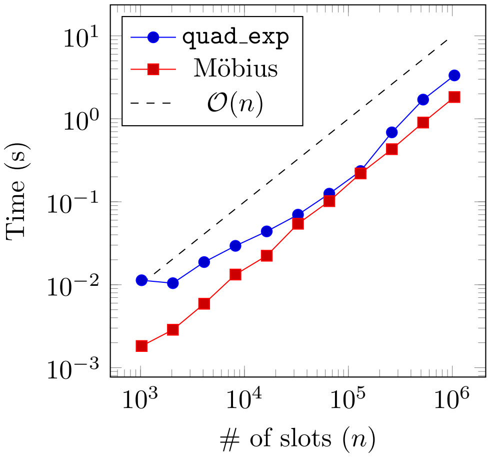
We have tested our implementation based on the quadrature scheme described in [1], and the timings needed to compute the measures as a function of the number of slots in the queue (that is, the size of the matrix Q) are reported in Figure 2.

The proposed approach has a linear complexity growth as the number n increases, as expected.

The accuracy requested was set to 10^{-8}. We note that, in this example, changing the number of available slots does not alter this measure in a distinguishable way: the states with a large index are very unlikely to be reached in a single unit of time, and therefore have a very low influence on the distribution $\pi(t)$ with $t = 1$.

6.2. Availability modeling for a telecommunication system

We consider an example taken from [29][Example 9.15], which describes a telecommunication switching system with fault detection / reconfiguration delay. This model describes n components which may fail independently, with a mean time to failure of $\frac{1}{\gamma}$. After failure of one component, this situation is



n	Time (s)
1,024	$1.13 \cdot 10^{-2}$
2,048	$1.04 \cdot 10^{-2}$
4,096	$1.87 \cdot 10^{-2}$
8,192	$2.94 \cdot 10^{-2}$
16,384	$4.39 \cdot 10^{-2}$
32,768	$6.99 \cdot 10^{-2}$
65,536	0.13
$1.31 \cdot 10^5$	0.23
$2.62 \cdot 10^5$	0.69
$5.24 \cdot 10^5$	1.7
$1.05 \cdot 10^6$	3.34

Figure 2: Time needed to approximate the average number of clients at time 1 depending on the total number of slots. The sizes range from 2^{10} to 2^{20}.

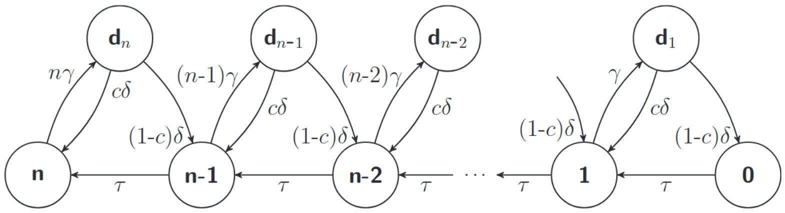


Figure 3: Pictorial description of the telecommunication switching system with fault detection / reconfiguration, modeled through a Markov chain. Figure taken from [29]

detected and the entire system switches to *detected mode*. If this happens, the component is repaired with a certain probability c (coverage factor), expected time of $\frac{1}{\delta}$, and the system came back to *normal mode*; otherwise, with probability $1 - c$, the component remains failed, the system switches to *normal mode* anyway, and the failed component is repaired with expected time $\frac{1}{\tau}$. The pictorial description of the system is reported in Figure 3, and the nonzero structure of the infinitesimal generator Q is reported in Figure 4. This Markov chain is irreducible, and we compute, fixed the interval $[0, t]$, the average time the system spends in detected mode from time 0 to time t.

Denoting with $\mathbf{d}$ the set of detected states, labeled as d_i for $i = 1, \ldots, n$ in Figure 3, this measure, rephrasing what already seen in Section 3.5, can be computed as

$$D(t) = \int_0^t \mathbb{E}\left[r_{X(\tau)}\right] \, d\tau, \qquad r = \mathbb{1}_{\mathbf{d}}.$$

In our test, we consider the interval of time $[0, 20]$, i.e., $t = 20$. The param-

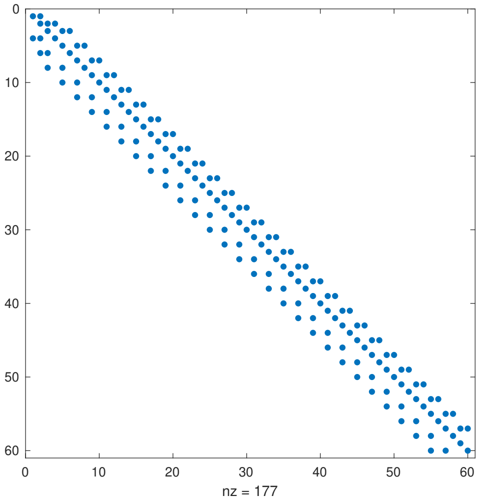


Figure 4: Non-zero structure of leading 60×60 minor of the infinitesimal generator Q for the system described in Section 6.2. The matrix Q is banded, and the structure is repeated along the diagonal.

eters are chosen as follows

$$c = 0.2, \qquad \delta = 0.5, \qquad \gamma = 0.95, \qquad \tau = 1.0.$$

In order to assess the scalability of our approach when the number of states grows, we consider large values of n (even though those may not be common for the particular situation of a telecommunication system). The number of states in the Markov chain can be shown to be $2n+1$. The computational time required to compute the two measures is reported in Figure 5 for different values of n ranging between 2^{10} and 2^{15}. We compare the timings with the cumulative solver included in Möbius 2.5 [10] that implement the uniformization method.

The timings shows that, in this case, the computational complexity on the solver bundled with Möbius seems to have a quadratic complexity in the number of states. This appears to be caused by an increasing number of iteration needed to reach convergence due to relevant differences among rates in the Markov chain, which causes stiffness in the underlying ODE. The Krylov approach, on the other hand, does not suffer this drawback.

6.3. Reliability model for communication system attacks

We consider the mobile cyber-physical system model presented in [20], describing a collection of communicating nodes which are subject to attacks. The original study is based on a real-world architecture: there are N mobile nodes, each node using sensors for localization and measuring anomaly phenomena, and the system comprises an imperfect intrusion/detection functionality distributed

Instantaneous availability

n	Möbius	`exp_phi`
2,049	0.8	$3.11 \cdot 10^{-2}$
4,097	2.84	$5.15 \cdot 10^{-2}$
8,193	11.54	$6.43 \cdot 10^{-2}$
16,385	50.05	0.1
32,769	247.92	0.13

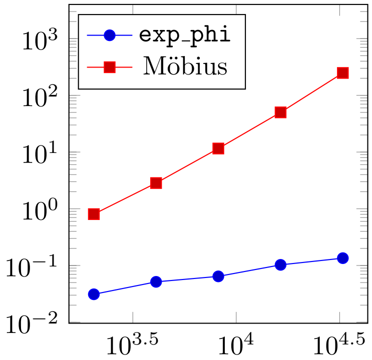


Figure 5: Timings for the computation of the cumulative measure in the telecommunication system, described in Section 6.2.

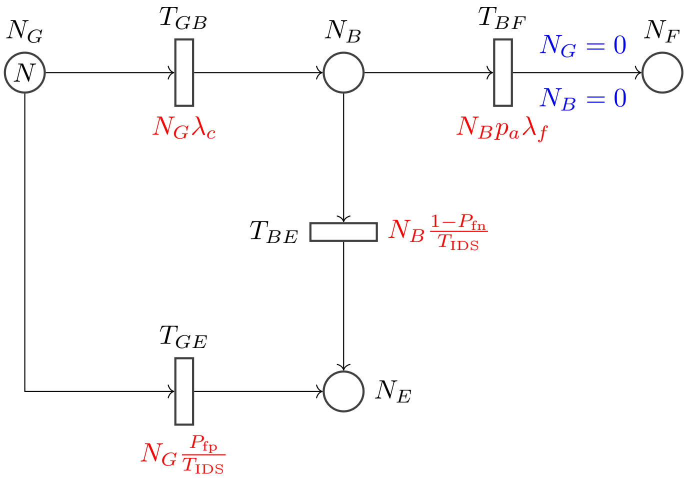


Figure 6: Attack model for the cyber-physical communication system described in Section 6.3. This model is a simplified version of the model presented in [20]. Places are represented as circles, transitions are represented as rectangles. Place and transition names are in black, transition rates are in red and actions performed whenever transition T_{BF} completes are in blue.

to all nodes for dealing with both intrusion and fault tolerance. This mechanism is based on a voting system. Here a simplified version is discussed, we refer the reader to [20] for further details on the intrusion/detection functionality and the complete system description.

The model considers a node capture which involves taking control of a *good* node by deceiving the authentication and turning it into a *bad* node that will be able to generate attacks within the system. The attackers primary objective is to cause impairment failure by performing persistent, random, or insidious attacks. At each instant of time, the number of good and bad nodes are indicated as N_G and N_B, respectively, and N_E is the number of *evicted* nodes, i.e., nodes that have been detected as bad ones by the intrusion/detection mechanism. At the beginning, all nodes are considered good, i.e., $N_G = N$. Only bad nodes can perform internal attacks, and whenever one of this attacks have success the entire system fails, switching the value of N_F from 0, *ok*, to 1, *failed*.

The model is expressed through the definition of the Stochastic Reward Net depicted in Figure 6.3, where places (circles) correspond to N_G, N_B, N_F, N_E, and determine the state of the system, and transitions (rectangles) define the behaviour of the attach model

- transition of a node from good to bad, called T_{GB}, represents the capture of a node by an attacker. The capture of a single node take place with rate λ_c, thus, being the capture of a node independent from the capture of other nodes, the rate of transition T_{GB} is $N_G\lambda_c$.
- transition of a node from bad to evicted, called T_{BE}, represents the correct detection of an attack. Calling P_{fn} the probability of intrusion/detection false negative, and T_{IDS} the period at which the intrusion/detection mechanism is exercised, the rate of T_{BE} is $N_B \frac{1-P_{\text{fn}}}{T_{\text{IDS}}}$.
- transition of a node from good to evicted, called T_{GE}, represent a false positive of the intrusion/detection mechanism. Calling P_{fp} the probability of intrusion/detection false positive, the rate of T_{GE} is $N_G \frac{P_{\text{fp}}}{T_{\text{IDS}}}$.
- transition of a the entire system from ok to failed, called T_{BF}. When a node is captured it will perform attacks with a probability p_a and the success of attacks from N_B compromised nodes has rate λ_f, thus the rate of T_{BF} is $N_B p_a \lambda_f$. At completion of transition T_{BF} the entire system fails and then both N_G and N_B are set to 0 so that the Stochastic Reward Net reach a (failed) absorbing state.

The graph whose vertexes are all the feasible combinations of values within places and arcs correspond to transitions forms the Markov chain under analysis. For instance, with $N = 3$ The Stochastic Reward Net of Figure 6.3 produces the Markov chain depicted in Figure 6.3, where the notation (n_G, n_B, n_E, n_F) means $N_G = n_G, N_B = n_B, N_E = n_E$ and $N_F = n_F$. The nonzero structure of the infinitesimal generator Q is reported in Figure 8. The parameters are chosen as follows

$$p_a = 0.7, \qquad P_{\text{fn}} = P_{\text{fp}} = 0.1, \qquad T_{\text{IDS}} = 15.0, \qquad \lambda_c = 0.1, \qquad \lambda_f = 0.2.$$

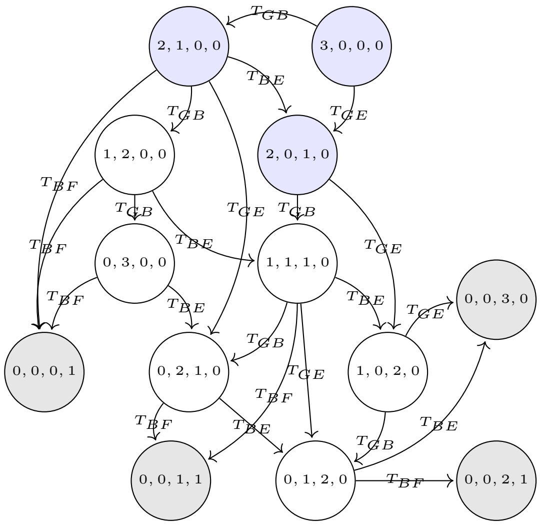


Figure 7: Markov chain produced by the Stochastic Reward Net depicted in Figure 6.3. The initial state is $(3, 0, 0, 0)$, the absorbing states are colored in gray, states such that $N_G \geq 2N_B$ are colored in blue.

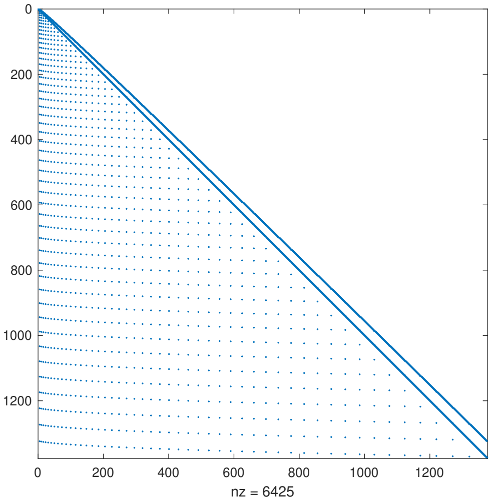


Figure 8: Nonzero structure of the infinitesimal generator Q for the system described in Section 6.3, where $n = 1376$.

Instantaneous availability

n	Möbius	`exp_phi`
1,376	$3.44 \cdot 10^{-2}$	$6.26 \cdot 10^{-2}$
5,251	0.21	0.54
11,626	1.25	0.6
20,501	5.34	0.83
31,876	14.54	1.24
45,751	34.52	1.62
62,126	62.76	2.39
81,001	110.43	3.33
$1.02 \cdot 10^5$	168.27	3.95
$1.26 \cdot 10^5$	258.45	7.64

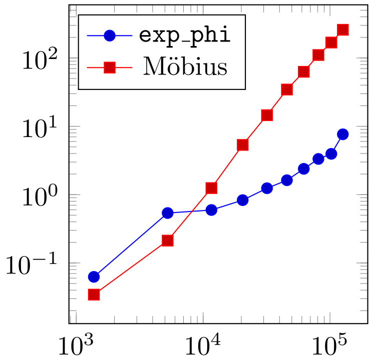


Figure 9: Timings for the computation of the cumulative measure in the security system, described in Section 6.3.

Being the intrusion/detection mechanism based on a voting system, the Byzantine fault model is selected to define the *security failure* of the system, i.e., the situation in which the system is working but there are not enough good nodes to obtain consensus when voting, that in our system means $N_G < 2N_B$. Thus, the cumulative measure of interest is

$$B_{\text{security}} = \int_0^t \mathbb{E}\left[r_{X(\tau)}\right] \, d\tau, \qquad r = \mathbb{1}_{\{N_G \geq 2N_B, N_F = 0\}}.$$

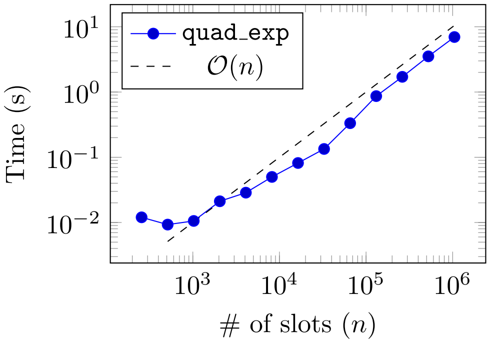
We have tested our implementation and measured the time required to compute this measure. The results are reported in Figure 9. Beside some overhead when dealing with small dimensions (and times below 0.01 seconds), the approach relying on the restarted Krylov method (labeled by `exp_phi`) is faster than the uniformization method included in Möbius.

6.4. Sensitivity analysis

As a last example, we consider the case of a sensitivity analysis. For simplicity, we consider once more the model of Section 6.1, and we assume to be interested in changing the parameter ρ_2. The only ingredient missing is computing the derivative of Q with respect to ρ_2, which in this case is simply given by the matrix

$$\frac{\partial Q}{\partial \rho_2} = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & & 0 \end{bmatrix}.$$

We have implemented the function computing the derivative of the measure following the approach described in Section 5, and the performance of the algorithm is reported in Figure 10, for the time $T = 1$. We see that also in this case



n	Time (s)
256	$1.2 \cdot 10^{-2}$
512	$9.28 \cdot 10^{-3}$
1,024	$1.06 \cdot 10^{-2}$
2,048	$2.12 \cdot 10^{-2}$
4,096	$2.87 \cdot 10^{-2}$
8,192	$4.99 \cdot 10^{-2}$
16,384	$8.17 \cdot 10^{-2}$
32,768	0.13
65,536	0.33
$1.31 \cdot 10^5$	0.87
$2.62 \cdot 10^5$	1.71
$5.24 \cdot 10^5$	3.52
$1.05 \cdot 10^6$	6.97

Figure 10: Time needed to approximate the sensitivity of the average number of clients at time 1 depending on the total number of slots. The sizes range from 2^8 to 2^{20}.

the scalability of the asymptotic cost of the algorithm is linear with respect to the number of states, as expected.

As demonstrated by the experiments, the Krylov based approach often outperforms the classical uniformization method. This could be explained by the fact that the Krylov method performs the number of restarts required to achieve a certain accuracy given a specific choice of π_0^T and r; in the case of the uniformization, instead, one has to choose a small timestep when dealing with stiff problems, ignoring the effect of the left and right vectors π_0 and r.

7. Conclusions

We have presented a novel point of view on the formulation of availability, reliability, and performability measures in the setting of Markov chains. The main contribution of this work is to provide a systematic way to rephrase these measures in terms of bilinear forms defined by appropriate matrix functions.

A dictionary translating the most common measures in the field of Markov modeling to the one of matrix functions has been described. We have proved that, leveraging the software available for the efficient computation of the action of $f(Q)$ on a vector, we can easily devise a machinery that evaluates these measures with the same or better performances that states of the art solvers, such as the one included in Möbius (which implements the uniformization method).

In particular, our solver seems to be more robust to unbalanced rates in the matrix and stiffness in general, which can make a dramatic difference in some cases, as showcased by our numerical experiments.

The new formulation allows to study measures' sensitivity by a new perspective, namely Frechét derivatives. This appears to be a promising reformulation that, along with providing efficient numerical procedures, might give interesting theoretical insights in the future.

We expect that this new setting will allow to devise efficient method tailored to the specific structure of Markov chains arising, for instance, from different high-level modeling languages.

Several problems remain open for further study. For instance, we have analyzed the use of Krylov methods with restarts, but the use of rational Krylov methods appear promising as well. Moreover, often the infinitesimal generator have particular structures induced by the high-level formalism used to model the Markov chain — as it has often been noticed in the literature — which we have not exploited here. These topics will be subject to future investigations.

Appendix A. Continuous time Markov chains and reward structures

Markov models are typically defined using high level formalism and then translated into CTMCs. Most often, the modeler is interested in extracting relevant information on the Markov chain, such as probability of breakdown (or, from the opposite perspective, of completing operations *without* breakdown). This kind of information is described abstractly using a *reward structure*, which is defined at the higher level — using the MRP language. These measures assess the performance and dependability of the system, and so are called *performability measures* [23].

A.1. Definition of models and measures

Given a CTMC, a natural question is if the limit for $t \to \infty$ of the probability distribution $\pi(t)$ exists, and whether it depends on the initial choice π_0. To characterize this behavior, we need to introduce the concepts of *irreducibility.*

Definition A.1. Let $X(t)$ be a Markov chain with infinitesimal generator Q, and consider the directed graph $\mathcal{G}$ with nodes the set of states of $X(t)$, and with an edge from i to j if and only if $Q_{ij} > 0$. We say that $X(t)$ is *irreducible* if for for every two states i, j there exists a path connecting i to j. We say that $X(t)$ is *reducible* if it is not *irreducible.*

We shall partition the Markov processes in two classes: the irreducible ones, called *transient*, which in the finite case are also *positive recurrent*, and the reducible ones, which are called *terminating.*

Intuitively, a Markov chain is irreducible if there is always a nonzero probability of jumping from i to j, possibly through some intermediate jumps. This property is sufficient to guarantee the existence and uniqueness of the *steady-state* vector π, the limit of $\pi(t)$ for $t \to \infty$.

From the linear algebra point of view, it is often useful to notice that a Markov process is reducible if and only if the matrix Q can be made block upper triangular by permuting the rows and column (with the same permutation).

We refer to the book [29] for a more detailed analysis on the classification of Markov chains, which we do not discuss further.

Theorem A.2. *Let $X(t)$ be an irreducible Markov chain with a finite number of states and infinitesimal generator Q. Then, there exists a unique positive vector π such that $\pi^T = \lim_{t\to\infty} \pi_0^T e^{tQ}$, and π does not depend on π_0. Moreover, π^T generates the left kernel of Q.*

A.2. Spectral properties of Q and the steady-state

As we already pointed out, unless the Markov chain is irreducible, the steady state probability might not be unique — and therefore depend on the specific choice of initial configuration π_0. The uniqueness can be characterized by considering the spectral properties of Q.

Lemma A.3. *Let Q be the infinitesimal generator of a continuous time Markov chain. Then, Q is singular, and has $\mathbb{1}$ as right eigenvector corresponding to the eigenvalue* 0*; if the Markov process is irreducible, then the left kernel is generated by π^T, the steady-state distribution.*

The matrix Q has another distinctive feature from the linear algebra point of view, which we will use repeatedly in what follows.

Lemma A.4. *Let Q be the infinitesimal generator of a Markov chain. Then, $-Q$ is an M-matrix, i.e., there exists a positive α such that*

$$M = \hat{M} - \alpha I,$$

with $\hat{M}$ being a non-negative matrix with spectral radius bounded by α.

Lemma A.4 implies, by a straightforward application of classical Gerschgorin theorems [30], that all the eigenvalues of Q are contained in the left half of the complex plane.

The spectral features of Q are tightly connected with the asymptotic behavior of the Markov chain. In particular, as we have pointed out in the previous section, a reducible process has a matrix Q that can be permuted to be block upper triangular. More precisely, one can reorder the entries as

$$\Pi Q \Pi^T = \begin{bmatrix} Q_{\mathbf{u}} & Q_{\mathbf{ud}} \\ Q_{\mathbf{du}} & Q_{\mathbf{d}} \end{bmatrix}, \qquad Q_{\mathbf{du}} = 0, \tag{A.1}$$

where Π is a permutation that lists the indices in $\mathbf{u}$ first, and then the ones in $\mathbf{d}$. The set u are the transient states of the process, whereas $\mathbf{d}$ contains the recurrent ones. The matrix $Q_{\mathbf{ud}}$ contains the probability rates of jumping from a state in $\mathbf{u}$ to a state in $\mathbf{d}$. The following characterization will be relevant in the following.

Lemma A.5. *Let Q be the infinitesimal generator of a terminating process, and Π the permutation identified in* (A.1)*. Then, $Q_{\mathbf{u}}$ is invertible.*

Proof. We claim that every row of $Q_{\mathbf{u}}$ has the sum of the off-diagonal elements strictly smaller than the modulus of the diagonal one. We assume that the matrix has been permuted already, so we may write without loss of generality:

$$\mathbf{u} = \{1, \dots, i'\}, \qquad \mathbf{d} = \{i'+1, \dots, n\}.$$

We have that, for every $i \leq i'$:

$$\sum_{j \leq i'} (Q_{\mathbf{u}})_{ij} + \sum_{i'+1 \leq j \leq n} (Q_{\mathbf{ud}})_{i,j-i'} = 0 \implies \sum_{j \leq i', j \neq i} (Q_{\mathbf{u}})_{ij} + (Q_{\mathbf{u}})_{ii} < 0.$$

Considering that the diagonal element has negative sign, and all the others are positive, we conclude that

$$|(Q_{\mathbf{u}})_{ii}| > \sum_{j \leq i', j \neq i} |(Q_{\mathbf{u}})_{ij}|,$$

and therefore 0 is not included in any of the Gerschgorin circles of $Q_{\mathbf{u}}$, and $Q_{\mathbf{u}}$ is invertible [30]. □

A.3. Available solution methods

Solving the Markov chain means computing the probability vector $\pi(t)$ at a given time t or, if the chain is irreducible, computing the steady-state probability vector π. As discussed in the following, often it is required to compute also $\int_0^t \pi_i(\tau) d\tau$ for some index i. We recall in this section the most well-known methods to tackle this task in the generic case (without particular assumption on the Markov chain).

A simple approach is the direct computation of the matrix exponential e^{tQ}, which in turns allows to obtain $\pi(t)^T = \pi_0^T e^{tQ}$. However, this is only feasible if the number of states is small, because the complexity is cubic in the number of states.

Another strategy is to compute the Laplace transform [29] of $\pi(t)$, denoted by $\bar{\pi}(s)$, solving the linear system $\bar{\pi}^T(s)\big(sI - Q\big) = \bar{\pi}_0^T$, obtained applying the Laplace transform to the Kolmogorov forward equation, and then anti-transform $\bar{\pi}(s)$ producing $\pi(t)$. The parameter s can be chosen so that $sI - Q$ is non-singular, the linear system can be solved exploiting favorable properties of Q, e.g., sparseness, and the known term $\bar{\pi}_0$ is often easy to compute because π_0 in dependability and performance models is highly structured. This method can be applied only to relatively small chains, being the anti-transform a costly operation, but can tackle relatively large t. Numerical integration [27] from zero to t of the Kolmogorov forward equation and/or of

$$\begin{cases} \dot{L}(t) &= L^T(t)Q + \pi_0^T, \\ L(0) &= 0, \end{cases}$$

where $L = \int_0^t \pi(\tau) d\tau$, is another alternative. Unfortunately, in dependability models the parameters can have different order of magnitudes, e.g., in a cyber-physical system the mean time to failure of an hardware component is considerably different from the mean time to failure of a software component, and similarly in performability models the performance-oriented and fault-related parameters can have different scalings. Thus, numerical integration is a good choice only if the chosen method, for the particular case under analysis, has been proved to be highly resilient to stiffness. For general chains and arbitrary t, the *uniformization method* [27] is commonly adopted by commercial level software tool, such as Möbius [10]. An heuristically chosen q such that $q > \max_i |Q_{ii}|$ allows to compute the truncate series expansion of $\pi(t)$ and $\int_0^t \pi(\tau)d\tau$ in terms of powers of $Q^* = (\frac{Q}{q} + I)$. In the numerical experiments, we will compare the performances of the approach proposed in this paper with the solver bundled with Möbius.

A.4. Matrix functions

Matrix functions are ubiquitous in applied mathematics, and appear in diverse applications. We refer the reader to [17] and the references therein for more detailed information. The definition of a matrix function can be given in different (equivalent) ways. Here we recall the one based on the Jordan form.

Definition A.6 (Matrix function)**.** Let A be a matrix with spectrum $\sigma(A) = \{\lambda_1, \dots, \lambda_n\}$, and $f(z)$ a function that is analytic on the spectrum of A. Let $J = V^{-1}AV$ be the Jordan form of A, with $J = J_1(\lambda_{j_1}) \oplus \dots \oplus J_k(\lambda_{j_k})$ being its decomposition in elementary Jordan blocks; we define the *matrix function* $f(A)$ as

$$f(A) = Vf(J)V^{-1}, \qquad f(J) = f(J_1) \oplus \dots \oplus f(J_k)$$

where for an $m \times m$ Jordan block we have:

$$f(J(\lambda)) = \begin{bmatrix} f(\lambda) & f'(\lambda) & \dots & f^{(m)}(\lambda) \\ & \ddots & \ddots & \vdots \\ & & \ddots & f'(\lambda) \\ & & & f(\lambda) \end{bmatrix}$$

Definition A.6 is rarely useful (directly) from the computational point of view. In most cases, computation of matrix functions is performed relying on the Schur form, on block diagonalization procedures, on contour integration, or on rational and polynomial approximation (we refer to [17] and the references therein for a comprehensive analysis of advantages and disadvantages of the different approaches).

Remark A.7. Note that, in order to apply Definition A.6, we do not necessarily need $f(z)$ to be analytic on the whole spectrum of A; we just need f to have derivatives of order m at every point corresponding to an eigenvalue with Jordan blocks of size at most $m + 1$.

Some examples of matrix functions that appear frequently in applied mathematics are the matrix exponential e^A, the inverse A^{-1} and the resolvents $(sI - A)^{-1}$, the square root $A^{\frac{1}{2}}$ and the matrix logarithm $\log(A)$.

Appendix B. Rephrasing the measures

This section aims to provide the building blocks that enables the translation of the measures from the Markov chain setting, where they are expressed as expected values of a random variable obtained as a function of $X(t)$, to the more computationally-friendly matrix function form.

Proof of Lemma 2.3. Recall that, by Definition 2.1, we have

$$M(t) = \int_0^t M_{\text{inst}}(\tau)\ d\tau = \pi_0^T \left(\int_0^t e^{\tau Q}\ d\tau \right) r,$$

by linearity of the integral. Assume, for simplicity, that Q is diagonalizable, and let $Q = VDV^{-1}$ be its eigendecomposition; we can write

$$M(t) = \pi_0^T V \left(\int_0^t e^{\tau D}\ d\tau \right) V^{-1} r = \pi_0^T V f(D) V^{-1} r, \quad f(z) = \int_0^t e^{\tau z}\ d\tau.$$

By direct integration we get $f(z) = \frac{e^{\tau z}-1}{z}$. Finally, using the relation $Vf(A)V^{-1} = f(VAV^{-1})$ we obtain the sought equality $M(t) = \pi_0^T f(Q) r$. The general statement follows by density of diagonalizable matrices, together with the continuity of $f(z)$. □

Remark B.1. We shall note that $t\varphi_1(tz)$ defined in Lemma 2.3, despite being defined piece-wise, is an analytic function, and is in fact defined by the power-series

$$t\varphi_1(tz) = t \cdot \left(1 + \frac{tz}{2} + \frac{(tz)^2}{3!} + \ldots + \frac{(tz)^j}{(j+1)!} + \ldots \right),$$

which is convergent for all $z \in \mathbb{C}$. Nevertheless, the formula $(e^{tz} - 1)/z$ cannot be used directly to evaluate the function at Q, because is not well-defined when $z = 0$, and this point is always included in the spectrum, since Q is singular. The notation $\varphi_1(z) = (e^z - 1)/z$ is often used in the description of exponential integrators. We refer to [16] and the references therein for more details.

A similar statement can be given also for the steady-state case. Nevertheless, to achieve this we need to consider a discontinuous function, and this can cause difficulties in the numerical use of such characterization.

Lemma B.2. *Let $X(t)$ a continuous irreducible Markov process with infinitesimal generator Q, and assume an initial distribution of probability π_0. The steady-state distribution π can be expressed as follows:*

$$\pi^T = \pi_0^T \delta(Q), \qquad \delta(z) = \begin{cases} 1 & \textit{if } z = 0 \\ 0 & \textit{otherwise} \end{cases}$$

Proof. We notice that, for any finite time t, $\pi(t) = e^{tQ}$. Since the spectrum of Q is contained in $\{\Re(z) < 0\} \cup \{0\}$, it is sufficient to check that $f_t(z) = e^{tz}$ converges to $\delta(z)$ as $t \to \infty$ on this set. It is clear that, for every $z \neq 0$ in the left half plane, we indeed have $f_t(z) \to 0$, and the claim follows noting that $f_t(0) = 1 = \delta(z)$ independently of t. □

Remark B.3. The fact that $\delta(z)$ is not analytic on the spectrum of Q is not an obstruction in the application of Definition A.6. In fact, since 0 is a simple eigenvalue, the definition is still applicable in view of Remark A.7.

Remark B.4. If the process is irreducible then the matrix function $\delta(Q)$ takes the form $\delta(Q) = \mathbb{1}\pi^T$ and therefore, since $\pi_0^T\mathbb{1} = 1$ for every probability distribution π_0, it is clear that the steady-state is independent of π_0.

References

[1] Afanasjew, M., Eiermann, M., Ernst, O.G., Güttel, S., 2008. Implementation of a restarted Krylov subspace method for the evaluation of matrix functions. Linear Algebra Appl. 429, 2293–2314.

[2] Al-Mohy, A.H., Higham, N.J., 2011. Computing the action of the matrix exponential, with an application to exponential integrators. SIAM J. Sci. Comput. 33, 488–511.

[3] Avizienis, A., Laprie, J.C., Randell, B., Landwehr, C., 2004. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secur. Comput. 1, 11–33. doi:`10.1109/TDSC.2004.2`.

[4] Balsamo, S., Onvural, R.O., Persone, V.D.N., 2001. Analysis of Queueing Networks with Blocking. Kluwer Academic Publishers, Norwell, MA, USA.

[5] Barbaresco, F., 2009. New foundation of radar doppler signal processing based on advanced differential geometry of symmetric spaces: Doppler matrix cfar and radar application, in: International Radar Conference.

[6] Benzi, M., Boito, P., 2014. Decay properties for functions of matrices over C^*-algebras. Linear Algebra Appl. 456, 174–198.

[7] Bini, D.A., Massei, S., Robol, L., 2017a. Efficient cyclic reduction for quasi-birth-death problems with rank structured blocks. Appl. Numer. Math. 116, 37–46.

[8] Bini, D.A., Massei, S., Robol, L., 2017b. On the decay of the off-diagonal singular values in cyclic reduction. Linear Algebra Appl. 519, 27–53.

[9] Crouzeix, M., Palencia, C., 2017. The numerical range is a $(1+\sqrt{2})$-spectral set. SIAM J. Matrix Anal. Appl. 38, 649–655.

[10] Deavours, D.D., Clark, G., Courtney, T., Daly, D., Derisavi, S., Doyle, J.M., Sanders, W.H., Webster, P.G., 2002. The Möbius framework and its implementation. IEEE Trans. on Softw. Eng. 28, 956–969.

[11] Demmel, J.W., 1997. Applied numerical linear algebra. volume 56. Siam.

[12] Estrada, E., Higham, D.J., 2010. Network properties revealed through matrix functions. SIAM Rev. 52, 696–714.

[13] Fenu, C., Martin, D., Reichel, L., Rodriguez, G., 2013. Network analysis via partial spectral factorization and Gauss quadrature. SIAM J. Sci. Comput. 35, A2046–A2068.

[14] Fletcher, P.T., Joshi, S., 2007. Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Process. 87, 250–262.

[15] Frommer, A., Güttel, S., Schweitzer, M., 2014. Efficient and stable Arnoldi restarts for matrix functions based on quadrature. SIAM J. Matrix Anal. Appl. 35, 661–683.

[16] Gander, M.J., Güttel, S., 2013. PARAEXP: a parallel integrator for linear initial-value problems. SIAM J. Sci. Comput. 35, C123–C142.

[17] Higham, N.J., 2008. Functions of matrices. Theory and computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

[18] Hillston, J., 1996. A Compositional Approach to Performance Modelling. Cambridge University Press, New York, NY, USA.

[19] Hochbruck, M., Lubich, C., Selhofer, H., 1998. Exponential integrators for large systems of differential equations. SIAM J. Sci. Comput. 19, 1552–1574.

[20] Martinez, J., Trivedi, K., Cheng, B., 2017. Efficient computation of the mean time to security failure in cyber physical systems, in: Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools, ICST, ICST, Brussels, Belgium. pp. 109–115. doi:10.4108/eai.25-10-2016.2266825.

[21] Massei, S., Robol, L., 2017. Decay bounds for the numerical quasiseparable preservation in matrix functions. Linear Algebra Appl. 516, 212–242.

[22] Meyer, J.F., 1980. On evaluating the performability of degradable computing systems. IEEE Transactions on computers , 720–731.

[23] Meyer, J.F., 1982. Closed-Form Solutions of Performability. IEEE Transactions on Computers C-31, 648–657. doi:10.1109/TC.1982.1676062.

[24] Plemmons, R.J., 1977. M-matrix characterizations. i—nonsingular m-matrices. Linear Algebra and its Applications 18, 175–188.

[25] Rathi, Y., Tannenbaum, A., Michailovich, O., 2007. Segmenting images on the tensor manifold, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 1–8.

[26] Reibman, A., Smith, R., Trivedi, K., 1989. Markov and Markov reward model transient analysis: An overview of numerical approaches. European Journal of Operational Research 40, 257 – 267. doi:https://doi.org/10.1016/0377-2217(89)90335-4.

[27] Reibman, A., Trivedi, K., 1989. Transient analysis of cumulative measures of Markov model behavior. Communications in Statistics. Stochastic Models 5, 683–710.

[28] Trivedi, K.S., 2002. Probability and Statistics with Reliability, Queuing and Computer Science Applications. 2nd ed., John Wiley and Sons Ltd., Chichester, UK.

[29] Trivedi, K.S., Bobbio, A., 2017. Reliability and Availability Engineering: Modeling, Analysis, and Applications. Cambridge University Press. doi:10.1017/9781316163047.

[30] Varga, R.S., 2010. Geršgorin and his circles. volume 36. Springer Science & Business Media.

[31] Yang, Q., Turner, I., Liu, F., Ilić, M., 2011. Novel numerical methods for solving the time-space fractional diffusion equation in two dimensions. SIAM J. Sci. Comp. 33, 1159–1180.

[32] Zhifeng, D., Fenghua, W., 2018. A generalized approach to sparse and stable portfolio optimization problem. Journal of Industrial & Management Optimization 14, 1651. doi:10.3934/jimo.2018025.