UNIVERSITÀ DEGLI STUDI FIRENZE

DISIA
DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

Analysis of probabilistic systems via generating functions and Padé approximation

Michele Boreale

DISIA WORKING PAPER
2016/10

Analysis of probabilistic systems via generating functions and Padé approximation*

Michele Boreale

Università di Firenze

Abstract

We investigate the use of generating functions in the analysis of discrete Markov chains. Generating functions are introduced as power series whose coefficients are certain hitting probabilities. Being able to compute such functions implies that the calculation of a number of quantities of interest, including absorption probabilities, expected hitting time and number of visits, and variances thereof, becomes straightforward. We show that it is often possible to recover this information, either exactly or within excellent approximation, via the construction of Padé approximations of the involved generating function. The presented algorithms are based on projective methods from linear algebra, which can be made to work with limited computational resources. In particular, only a black-box, on-the-fly access to the transition function is presupposed, and the necessity of storing the whole model is eliminated. A few numerical experiments conducted with this technique give encouraging results.

Keywords: Probabilistic systems, Markov chains, generating functions, Padé approximation.

1 Introduction

Our goal is to understand if the concept of generating function [19] can play a useful role in the analysis of Markov chains. In the present paper, we focus on the reachability properties of time-homogeneous, finite Markov chains. The generating function of such a system is a power series in the variable z, $g(z) = \sum_{j\geq 0} a_j z^j$, whose coefficients, or *moments*, a_j are just the probabilities of hitting a state of interest exactly at time $j = 0, 1, 2, \cdots$. With $g(z)$, a whole host of information about the system is packed into a single mathematical object, including: the probability of the event itself - which is of course $g(1)$ - and various statistics, such as the expected hitting time and its variance. We will demonstrate that, by building a rational representation of $g(z)$, in a number of interesting situations it is possible to extract this information, either exactly or within excellent approximation, using limited computational resources. These limitations are mainly the fact that one can access the system's transition relation only in a black-box, on-the-fly[1] fashion, and can only store a small portion of its state space at time. We give a more detailed account of our approach and of our paper below.

We first introduce and motivate the system's generating function $g(z)$, then establish some of its important properties, like its radius convergence and its rationality (Section 2). We then show (Section 3) that, via *Padé approximants* [5], an *exact* rational representation of $g(z)$ can be recovered from the knowledge of its first $2N$ moments, where N is the number of system's states. These moments can in principle be computed relying solely on a black-box, on-the-fly access to the transition relation, and do not require storing the entire model. Yet, limited resources imply that one is often forced to consider approximations. We argue (Section 4) that polynomial approximations, derived from truncating $g(z)$, may not be a good idea, and that rational ones should rather be preferred. This is especially true in the presence of clusters of states that are nearly uncoupled with other states in the chain. We then discuss a method to compute one such approximation effectively (Section 5).

*Extended version of [7]. Author's address: Michele Boreale, Università di Firenze, Dipartimento di Statistica, Informatica, Applicazioni (DiSIA) "G. Parenti", Viale Morgagni 65, I-50134 Firenze, Italy. E-mail: michele.boreale@unifi.it. Work partially supported by by MIUR funded project CINA.

[1] That is, via a function that given a state returns the list of its successors together with their probabilities.

The basic idea here is to view the matrix P that represents the transition relation as a linear application on $\mathbb{R}^N$; then to take its projection $\hat{P}$ onto a small, m-dimensional subspace $\mathcal{K}_m$, with $m \ll N$. The generating function of the projected application, $\hat{g}(z)$, is a rational *Padé-type* approximation of the original $g(z)$. Accuracy is often very good already for small m: this somewhat surprising effectiveness is a consequence of the tendency of $\hat{P}$'s eigenvalues to be excellent approximations of P's ones. We show that the Arnoldi algorithm [3, 18] can be used to effectively compute $\hat{g}(z)$, in a way that is compatible with an on-the-fly access to the transition relation and with limited computational resources. Notably, transitions need not be stored at all with this method. Error control and steady-state distributions are also discussed. We then present a few numerical experiments that have been conducted with a preliminary Matlab implementation of this idea (Section 6), and which give very encouraging results. For comparison, the results obtained on the same systems with a state-of-the-art probabilistic model checker are also reported. We conclude the paper with a discussion of future venues of research and related work (Section 7). All the proofs, some numerical examples and additional technical material have been confined to a separate appendix.

To sum up, we make the following main contributions: (1) we demonstrate the usefulness of the concept of generating function in the analysis of discrete Markov chains; (2) we show that, via Padé approximants, $g(z)$ can be computed without having to store a complete representation of the system; (3) we show that a Padé-type rational approximation $\hat{g}(z)$ can be computed with even more limited computational resources, and often yields an excellent approximate analysis of the original system.

2 The system generating function $g(z)$

Consider a time-homogeneous Markov chain $\{X_j\}_{j\geq 0}$ over a finite set of states $\mathcal{S} = \{1, ..., N\}$ with $N > 0$ and initial state $X_0 = 1$. To avoid uninteresting special cases, we will assume that all states of the chain are reachable from 1 (but need not assume the vice-versa, so the chain might well be reducible.) We want to study the event corresponding to reaching a (typically, 'bad') state $s_{bad} = N$, that is $Reach \triangleq \{X_j = N \text{ for some } j \geq 0\}$ and denote by $p_{reach} \triangleq \Pr(Reach)$ the probability of this event. Without loss of generality, we will assume that state N is absorbing that is $\Pr(X_{j+1} = N | X_j = N) = 1$. Later on in this section we will also consider another type of statistics, concerning the number of visits, where we will not assume N is absorbing. We will also be interested in statistics concerning the *hitting time* random variable, defined thus

$$T \triangleq \inf\{j \geq 0 : X_j = N\}.$$

Let us define the j-th ($j \geq 0$) *moment* of the system as the probability of hitting state N for the first time exactly at time j, that is

$$a_j \triangleq \Pr(X_j = N \text{ and } X_i \neq N \text{ for } i < j). \tag{1}$$

Clearly, the probability of eventually reaching N is just the sum of the moments: $p_{reach} = \sum_{j\geq 0} a_j$. Our main object of study is defined below.

Definition 1 (Generating function) *The generating function of the system is the power series in the complex variable z*

$$g(z) \triangleq \sum_{j\geq 0} a_j z^j. \tag{2}$$

Note that the $g(z)/p_{reach}$ is just the probability generating function of the random variable T conditioned on the event *Reach*. As such, with $g(z)$ a whole host of information about T and its moments is packed into a single mathematical object. For instance, indicating with $g', g'', ...$ the derivatives of g, easy calculations show that (provided the mentioned quantities are all defined)

$$p_{reach} = g(1) \qquad E[T|Reach] = g'(1)/g(1) \qquad \text{var}[T|Reach] = \left(g''(1) + g'(1)\right)/g(1) - \left(g'(1)\right)/g(1))^2 . \tag{3}$$

More generally, information on higher moments of T can be extracted using the identity $E[X(X-1)\cdots(X-k)|Reach] = g^{(k+1)}(1)/g(1)$, although usually the first two moments are enough for a satisfactory analysis of the system. In practice, we will be able to extract this information only provided we are able to build an efficient representation of $g(z)$. As a first step towards this, we shall see in a moment that $g(z)$ can be represented as a

rational function, that is, as the ratio of two polynomials in z. At this point it is convenient to introduce some notation.

Notation In the rest of the paper, some basic knowledge of linear algebra is presupposed. We let P denote the $N \times N$ stochastic matrix that defines the transition function of the chain. Departing from the usual convention, we will work with *column*-stochastic matrices, that is we let the element of row i and column j of P, denoted by p_{ij}, be $\Pr(X_{t+1} = i | X_t = j)$. In what follows, vectors are considered as column-vectors; in particular, e_i denotes the i-th canonical column vector of $\mathbb{R}^N$. So the vector $P^i e_1$ is just the probability distribution of the variable X_i of the chain. A vector is stochastic if its components are nonnegative and sum to 1. For a matrix or vector A, we let A^T denote its transpose. I_k will denote the $k \times k$ identity matrix; the index k will be omitted when clear from the context. A rational function in z of type $[h, k]$, for $k \geq 0$ and $h \geq 0$ or $h = -\infty$, is a ratio of two polynomials in z, $r(z)/t(z)$, such that $\deg(r) \leq h$ and $\deg(t) \leq k$. We will let z range over complex numbers and x on reals.

In the rest of the paper, we let

$$\tilde{e}_1 \triangleq (P - I)e_1 \,.$$

Note that $e_N^T P^j e_1$ is just the probability of being in state N at time j. Exploiting the fact that N is absorbing, it is easy to see that, for $j \geq 1$

$$a_j \;\; = \;\; e_N^T P^j e_1 - e_N^T P^{j-1} e_1 \;\; = \;\; e_N^T P^{j-1} \tilde{e}_1 \,. \tag{4}$$

Ignoring for a moment issues of convergence and singularity, we can then reason as follows. We note that the following equality can be readily checked

$$(I - zP)(I + zP^1 + z^2 P^2 + \cdots) = I$$

which implies that $(I - zP)^{-1} = (I + zP^1 + z^2 P^2 + \cdots)$. We then can write

$$\begin{aligned} g(z) = a_0 + \sum_{j \geq 1} a_j z^j \qquad & = a_0 + \sum_{j \geq 1} e_N^T z^j P^{j-1} \tilde{e}_1 \\ = a_0 + z \cdot e_N^T (\sum_{j \geq 0} z^j P^j) \tilde{e}_1 \quad & = a_0 + z \cdot e_N^T (I - zP)^{-1} \tilde{e}_1 \\ = a_0 + \frac{z \cdot e_N^T \mathrm{Adj}(I - zP) \tilde{e}_1}{\det(I - zP)} & \end{aligned} \tag{5}$$

where, in the last step, we have exploited Cramer's rule for the computation of the inverse (recall that $\mathrm{Adj}(A)$ denotes the *adjoint matrix* of a matrix A.) In the last expression, the denominator and numerator of the fraction are polynomials in z. This shows that $g(z)$ is a rational function. This informal reasoning can be made into a rigorous proof – a nontrivial point of which is related to the singularity of the matrix $(I - zP)$ at $z = 1$. Another important point is where the power series $g(z)$ is defined, that is, what is its radius of convergence. The the next theorem records these facts about $g(z)$.

Theorem 1 (convergence and rationality of g) *There is a real $R > 1$ such that the power series $g(z)$ in (2) converges for all $|z| < R$. Moreover, there is a rational function $r(z)/t(z)$ such that for all such z's*

$$\textbf{(a)} \;\; t(z) \neq 0 \qquad \textbf{(b)} \;\; \deg(r), \deg(t) \leq N - 1 \qquad \textbf{(c)} \;\; g(z) = r(z)/t(z) \,. \tag{6}$$

The proof of the above theorem provides us also with an explicit expression for $g(z)$, that is (5). In what follows, $(I - zP)^{-1}$ will be used as an abbreviation for the matrix of rational expressions $\frac{\mathrm{Adj}(I - zP)}{\det(I - zP)}$, where it is understood that common factors are canceled out. Concerning this expression, note that $\det(I - zP) = z^N \det((1/z)I - P)$ is just the characteristic polynomial of P with coefficients reversed. That is, the relation between the characteristic polynomial and our $\det(I - zP)$ is as follows

$$\begin{aligned} \det(zI - P) &= \beta_N z^N + \beta_{N-1} z^{N-1} + \cdots + \beta_0 \\ \det(I - zP) &= \beta_N + \beta_{N-1} z + \cdots + \beta_0 z^N \,. \end{aligned}$$

(In passing, note that $\beta_N = 1$ and $\beta_0 = -\det(P)$.) In particular, from $\det(I - zP) = z^N \det((1/z)I - P)$ it is clear that the roots of the polynomial $\det(I - zP)$ are just the reciprocals of the nonzero roots of P: that is, the reciprocals of the nonzero eigenvalues of P. This fact can be exploited to give more precise information about R. We record these facts below.

Corollary 1 *There is $R > 1$ such that for $|z| < R$ and for $\tilde{e}_1 = (P - I)e_1$*

$$g(z) \;\; = \;\; a_0 + z \cdot e_N^T (I - zP)^{-1} \tilde{e}_1 \,. \tag{7}$$

In particular, $g(z)$ has a radius of convergence either $R = |1/\lambda|$ for some eigenvalue $0 < |\lambda| < 1$ of P, or $R = +\infty$.

Let us now drop the assumption that N is absorbing. We are interested in counting the visits to state N, starting from $X_0 = 1$. To this purpose we introduce a different generating function: $f(z) \stackrel{\triangle}{=} \sum_{j\geq 0} c_j z^j$, where $c_j \stackrel{\triangle}{=} \Pr(X_j = N) = e_N^T P^j e_1$. By definition, $f(1)$ is the expected number of visits of the chain to state N. Recall that $f(1) < +\infty$ iff N is transient. By paralleling the above development for $g(z)$, we can prove the following.

Theorem 2 *Let N be transient. The power series $f(z)$ has radius of convergence $R > 1$. Moreover, for $|z| < R$, one has $f(z) = e_N^T (I - zP)^{-1} e_1$. The expression on the right of the last equality gives rise to a rational function $r(z)/t(z)$ of type $[N-1, N]$ such that $t(z) \neq 0$ for $|z| < R$.*

From now on, we will consider $g(z)$ only; statements and proofs for $f(z)$ can be obtained by obvious modifications. All the quantities of interest about the system, (3), will be easy to compute, provided we can recover the rational representation $r(z)/t(z)$ of $g(z)$ promised by Theorem 1. The expression provided by Corollary 1 can be useful for small values of N and provided one knows P explicitly. Here is a small example to illustrate.

Example 1 We consider a chain with $N \geq 3$ states 1,2,..., N, where, for $1 \leq i \leq N-3$ and a (small) $0 < \delta < 1$, there is a transition from i to 1 with probability $1 - \delta/i$, and to each of $i+1, N-1, N$ with probability $\delta/3i$; for $i = N-2$, there is a transition from i to 1 with probability $1 - \delta/i$, and to each of $N-1, N$ with probability $\delta/2i$; $N-1$ and N are absorbing. For reasons that will become evident later on, we call this chain $Nasty(N, \delta)$.

$$P = \begin{bmatrix} 1-\delta & 1-\delta/2 & 1-\delta/3 & 1-\delta/4 & 0 & 0 \\ \delta/3 & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta/6 & 0 & 0 & 0 & 0 \\ 0 & 0 & \delta/9 & 0 & 0 & 0 \\ \delta/3 & \delta/6 & \delta/9 & \delta/8 & 1 & 0 \\ \delta/3 & \delta/6 & \delta/9 & \delta/8 & 0 & 1 \end{bmatrix}.$$

The transition matrix P of $Nasty(6, \delta)$ is given above (recall that we work with column-stochastic matrices.) From symmetry considerations, it is clear that the probability of reaching either of the two absorbing states is $1/2$, thus $p_{reach} = 1/2$. Let us check this out via $g(z)$. With the help of a computer algebra system, we apply (7) and, taking into account that $a_0 = 0$, find

$$g(z) = \frac{1}{2} \frac{\delta^4 z^4 + 8\delta^3 z^3 + 72\delta^2 z^2 + 432\delta z}{(\delta^4 - 4\delta^3) z^4 + (12\delta^3 - 36\delta^2) z^3 + (108\delta^2 - 216\delta) z^2 + 648(\delta - 1) z + 648}.$$

When evaluated at $z = 1$ we get

$$g(1) = p_{reach} = \frac{1}{2}.$$

By differentiating the above expression of $g(z)$ and then evaluating the resulting expression at $z = 1$, we get

$$g'(1) = 2 \frac{\delta^3 + 9\delta^2 + 54\delta + 162}{\delta^4 + 8\delta^3 + 72\delta^2 + 432\delta}.$$

which, can be evaluated for instance at $\delta = 10^{-3}$ to compute

$$E[T|Reach] = g(1)^{-1} g'(1) \approx 1500.25.$$

Computing an expression for $g(z)$ based on a direct application of Corollary 1 requires the explicit knowledge of the matrix P. Moreover, the computation relies on costly symbolic operations involving matrices whose entries are rational functions in z, rather than scalars. For these reasons, this method can only be practical for small values of N. The next section explains how to numerically calculate a rational representation of $g(z)$ out of the first $2N$ moments of the system, without having to know P explicitly.

3 Exact reconstruction of $g(z)$

We first review Padé approximants and then explain how to employ them to exactly reconstruct $g(z)$. The exposition of Padé approximants in this section is standard. For an in depth treatment, see e.g. [5]. Let $f(z) = \sum_{i\geq 0} c_i z^i$ be a generic a power series in the complex variable z, with a nonzero radius of convergence. For any $n \geq 0$, we let the truncation of f at the n-th term be the polynomial $f_n(z) \stackrel{\triangle}{=} \sum_{i=0}^{n} c_i z^i$. Let us indicate by

$o(z^k)$ a generic power series divisible by z^k. Given two power series $f(z)$ and $d(z)$ we write $f(z) = d(z) \bmod z^{n+1}$ iff $f_n = d_n$, or, in other words, if $f(z) - d(z) = o(z^{n+1})$. Polynomials are of course considered as power series with only finitely many nonzero coefficients.

Definition 2 (Padé approximants) *Let $f(z) = \sum_{i\geq 0} c_i z^i$ be a power series in the complex variable z with a nonzero radius of convergence. Given integers $h, k \geq 0$, we say a pair of polynomials in z, say (r, t), is a* $[h, k]$-Padé approximant of $f(z)$ *if the following holds true, where $n = h + k$.*

$$\textbf{(a)}\ \ z \not| t(z) \quad \textbf{(b)}\ \ \deg(r) \leq h \text{ and } \deg(t) \leq k \quad \textbf{(c)}\ \ f(z)t(z) = r(z) \bmod z^{n+1}\,. \tag{8}$$

Seen as a *real* function $r(x)/t(x)$, a Padé approximant is a rational approximation of the function $f(x)$, up to and including the term of degree n of its Taylor of expansion. To see this, first note that equation (8)(c) is equivalent to saying that there exists a power series $k(x)$ such that

$$f(x)t(x) \quad = \quad r(x) + k(x)x^{n+1}\,.$$

As $t(0) \neq 0$, the last equation is equivalent to saying that, in a neighborhood of 0

$$f(x) \quad = \quad \frac{r(x)}{t(x)} + \frac{k(x)}{t(x)}x^{n+1}\,.$$

This equation is equivalent to saying that the Taylor expansion of $\frac{r(x)}{t(x)}$ at $x = 0$, truncated at the n-th term, coincides with[2] $f_n(x)$, that is

$$\frac{r(x)}{t(x)} \quad = \quad \sum_{i=0}^{n} c_i x^i \ + \ o(x^{n+1})\,.$$

In case f is itself a rational function, Padé approximants provide us with a method to actually find an *exact* representation of it, given only sufficiently many coefficients c_i, as we will see shortly. We first state a result about uniqueness of Padé approximants; its proof follows from easy manipulations on rational functions (or see [5, Th.1.1].)

Proposition 1 *Let (r, t) and (p, q) be two $[h, k]$-Padé approximants of f. Then they are the same as a function, in the sense that $r(z)/t(z) = p(z)/q(z)$.*

Given any power series $f(z)$, it is possible to compute a $[h, k]$-Padé approximant of it as follows. Here we assume for simplicity that $h \leq k$; the case $h > k$ does not interest us, and can be anyway treated with minor notational changes. Also, in view of condition (8)(a) of the definition of Padé approximant, without loss of generality we will restrict ourselves to the case where $t(0) = 1$, that is, the constant coefficient of t is always taken to be 1.

Assume $n + 1$ ($n = h + k$) coefficients $c_0, c_1, ..., c_n$ of f are given. Arrange the coefficients from h through $h + k - 1 = n - 1$ to form a $k \times k$ matrix C as described on the right, where $c_l \stackrel{\triangle}{=} 0$ for indices $l < 0$. Let $r(z) = \alpha_h z^h + \cdots + \alpha_0$ and $t(z) = \beta_k z^k + \cdots + \beta_1 z + 1$ be two polynomials, for generic vectors of coefficients $\alpha = (\alpha_0, ..., \alpha_h)^T$ and $\beta = (\beta_1, ..., \beta_k)^T$.

$$C \quad = \quad \begin{bmatrix} c_h & c_{h-1} & \cdots & c_{h-k+1} \\ c_{h+1} & c_h & \cdots & c_{h-k+2} \\ & & \vdots & \\ c_{h+k-1} & c_{h+k-2} & \cdots & c_h \end{bmatrix}.$$

Assume (r, t) is a $[h, k]$-Padé approximant of f. Then we can equate coefficients of like powers on the left- and right-hand side of (8)(c). In particular, coefficients from $h + 1$ through $n = h + k$ are 0 on the right, thus coefficients on the left must satisfy the following, for $\gamma \stackrel{\triangle}{=} (-c_{h+1}, ..., -c_{h+k})^T$:

$$C\beta \quad = \quad \gamma\,. \tag{9}$$

On the other hand, assume (9), seen as a system of equations in the unknowns β, has a solution. Then by taking α given by:

$$\alpha = \tilde{C}\beta' \tag{10}$$

where $\beta' = (1, \beta_1, ..., \beta_h)^T$ and $\tilde{C}$ is the $(h + 1) \times (h + 1)$ lower-triangular matrix $\tilde{C}$ given on the right, we see that (8)(c) is satisfied by (r, t), as well as, of course, (8)(a) and (8)(b).

$$\tilde{C} = \begin{bmatrix} c_0 & & & & \\ c_1 & c_0 & & & \\ & & \vdots & & \\ c_h & c_{h-1} & \cdots & c_1 & c_0 \end{bmatrix}.$$

[2] To see this, observe that, for $0 \leq j \leq n$, by equating the j-th derivatives of the left and right hand side of (8)(c), one obtains that $f^{(j)}(x) = (\frac{r(x)}{t(x)})^{(j)} + o(x^{n-j+1})$, so that $c_j = \frac{f^{(j)}(0)}{j!} = \frac{1}{j!}(\frac{r(0)}{t(0)})^{(j)}$. Also note that the Taylor expansion of $\frac{r(x)}{t(x)}$ at $x = 0$ exists, as $t(0) \neq 0$.

Therefore (r, t), as given by α and β, is a $[h, k]$-Padé approximant of f. In other words, we have shown the following.

Proposition 2 *A $[h, k]$-Padé approximant for f exists if and only if the system of equations* (9) *in the unknowns β has a solution. If it exists, the coefficients β and α for t and r are given, respectively, by a solution of* (9) *and by* (10).

Note that the procedure outlined above to reconstruct a $[h, k]$-Padé approximant takes $O(k^3 + h^2) = O(n^3)$ operations and $O(n^2)$ storage. Let us now come back to our Markov chain. Theorem 1 and Proposition 1 ensure that $g(z)$ coincides, as a function, with its $[N-1, N-1]$-Padé approximant. Using the above outlined method, one can reconstruct the rational form of $g(z)$, provided one knows (an upper bound on) N and the coefficients $a_0, ..., a_{2N-2}$. The latter can in principle be computed by a power iteration method: $a_0 = e_N^T e_1$ and $a_i = e_N^T P^{i-1} \tilde{e}_1$ for $i \geq 1$. For this, it is sufficient to obtain a black-box access to the function $u \mapsto Pu$, which is compatible with an on-the-fly implementation. A numerical example illustrating the method is reported in Section C.

While dispensing with symbolic computations and the explicit knowledge of P, the method described in this section still suffers from drawbacks that confine its application to small-to-moderate values of N. Indeed, the solution of the linear system (9) has a time complexity of $O(N^3)$. Although this can be improved using techniques from numerical linear algebra (in fact, $O(N \log^2 N)$ algorithms exist, see e.g. [12]), the real problem here is that the explicit computation of $2N$ moments a_j is in practice very costly and numerically instable, and should be avoided. Moreover, it is clear that a consistent gain in efficiency can only be obtained by accepting some degree of approximation in the computed solution. The next two sections discuss how to achieve this without sacrificing accuracy too much.

4 Discussion: approximating $g(z)$

Suppose that, possibly due to limited computational resources, we have access only to a limited number, say $m + 1$, of g's moments a_i. Or, more generally, we can access the transition relation of the Markov chain - the mapping $u \mapsto Pu$ - only a limited number m of times. Typically, we can only afford $m \ll N$. The resulting information may be not sufficient to recover an exact representation of $g(z)$. Then the problem becomes finding a a good approximating function $\hat{g}(z)$ of $g(z)$. "Good" here means at least that $g(z) - \hat{g}(z) = o(z^{m+1})$; but, even more than that, $\hat{g}(z)$ should approximate well $g(z)$ near $z = 1$, as we are mainly interested in evaluating $g(1), g'(1)$ and so on, see (3).

The first, obvious attempt is to consider a polynomial approximation

$$\hat{g}(z) \quad = \quad g_m(z) = \sum_{i=0}^{m} a_i z^i \,. \tag{11}$$

This is the truncation of $g(z)$ at the term of degree m. Unfortunately, such a $\hat{g}(z)$ might be a very bad approximation of $g(z)$. The reason is that the rational representation $r(z)/t(z)$ of $g(z)$ may have a pole near[3] $z = 1$: that is, there can be a $z_0 \in \mathbb{C}$ such that $|z_0 - 1| \approx 0$ and $\lim_{z \to z_0} |r(z)/t(z)| = +\infty$. Then, as z approaches 1 from its convergence zone, $g(z)$ becomes extremely fast growing, and essentially impossible to approximate by means of a polynomial function, as polynomials have no finite poles.

As stated by Corollary 1, the pole of smallest modulus of $g(z)$, which determines its radius of convergence R, is of the form $z_0 = 1/\lambda$, for some subdominant eigenvalue λ of P, that is an eigenvalue with $|\lambda| < 1$. If 1 is "badly separated" from λ, that is if $|\lambda - 1| \approx 0$, the truncated sums $\sum_{i \leq m} a_i$ will converge very slowly to p_{reach}, as m grows. In this respect, a rational approximation

$$\hat{g}(z) \quad = \quad \frac{\hat{r}(z)}{\hat{t}(z)} \tag{12}$$

can perform much better. Indeed, $\hat{t}(z)$ can be chosen so as to have a root near z_0. This in essence is what Padé approximation achieves. When building a $[h, k]$-Padé approximant with $h + k \leq m$, the same amount of information used to build (11) - the first $m + 1$ moments of $g(z)$ - is used to "guess" an approximation of z_0, that becomes a root of $\hat{t}(z)$ (this aspect will be further discussed in the next section, see Remark 1.) The benefit of rational over polynomial approximation is qualitatively illustrated by the plots in Fig. 1.

[3] Note that the rational function $r(z)/t(z)$, while coinciding with $g(z)$ within the disk $|z| < R$, will also be defined outside this disk.

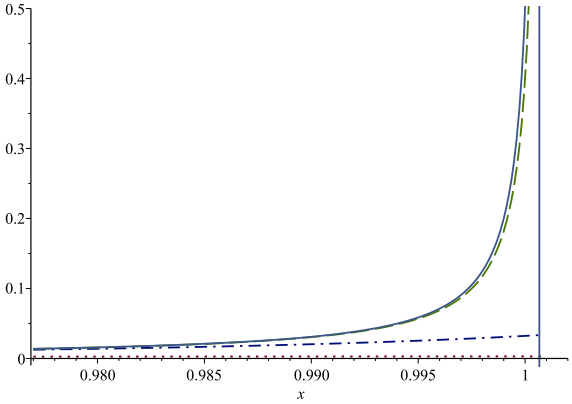

Figure 1: Plots of $g_{10}(x)$ (dotted), of $g_{100}(x)$ (dash-dotted), of the $[1,1]$-Padé approximant $\hat{g}(x) = 2x/(-5995x + 6000)$ (dashed) and of $g(x)$ (solid), for $Nasty(6, 10^{-3})$, near $x = 1$. Here $g(z)$ has a pole at $z = 1/\lambda$ with $\lambda \approx 0.9993$.

It is well-known that that the bad separation phenomenon (subdominant eigenvalues close to 1) occurs if there is a cluster of states that are strongly coupled with one another, but nearly uncoupled with other states in the chain, like $1, ..., n-2$ in $Nasty(n, \delta)$. For the reader's convenience a brief discussion of this aspect is reported in Appendix B.

In the next section we explore an effective way of building rational approximations of the form (12). The resulting expressions are known as *Padé-type approximant*, a weak form of Padé approximation.

5 Approximation of $g(z)$ via a projection method

The general idea of a projection method is as follows. Consider P as a linear map acting on the N-dimensional space $\mathbb{R}^N$. We identify a m-dimensional subspace, $\mathcal{K}_m$, and then consider the projection of P onto this space, say $\hat{P}$: this is our low-dimensional approximation of the original system. Here, $m \ll N$: practically m will be of the order of tens or hundreds. Then we essentially repeat the construction of Section 2, but with P replaced by $\hat{P}$, and extract a low-dimensional rational representation of its generating function, $\hat{g}(z)$. The space $\mathcal{K}_m$ will be constructed as a subset of the state vectors reachable from $\tilde{e}_1$ via P: this type of space is known in the linear algebra literature as Krylov space. There is an efficient and numerically stable method to build $\mathcal{K}_m$ and $\hat{P}$, known as as the Arnoldi method. We will first adapt this method to our purposes. Then discuss computation of steady state distributions and error control.

5.1 Main algorithm

Formally, consider an integer $m \geq 1$ and the *Krylov subspace* of $\mathbb{R}^N$

$$\mathcal{K}_m(P, \tilde{e}_1) \stackrel{\triangle}{=} \text{span}\{\tilde{e}_1, P\tilde{e}_1, P^2\tilde{e}_1, ..., P^{m-1}\tilde{e}_1\}$$

abbreviated as $\mathcal{K}_m$ in what follows. Now take any orthonormal basis of $\mathcal{K}_m$ and arrange the corresponding column vectors into a $N \times m$ matrix, $V_m = [v_1, ..., v_m]$. Note that orthonormality means that $V_m^T V_m = I_m$. We can consider the projection of P, seen as an application $\mathbb{R}^N \to \mathbb{R}^N$, onto $\mathcal{K}_m$. The representation of this application restricted to $\mathcal{K}_m$, in the basis V_m, is given by the $m \times m$ matrix

$$H_m \quad = \quad V_m^T P V_m \,. \tag{13}$$

(The matrix H_m will play the role played by $\hat{P}$ in the above informal description.) Note that if m is large enough then $\mathcal{K}_m$ will be a P-invariant subspace of $\mathbb{R}^N$, that is $P\mathcal{K}_m \subseteq \mathcal{K}_m$[4].

Theorem 3 *Let $m \geq 1$. Consider the function, defined in a neighborhood of the origin*

$$\hat{g}(z) \stackrel{\triangle}{=} a_0 + z \cdot (e_N^T V_m)(I_m - zH_m)^{-1}(V_m^T \tilde{e}_1) \,. \tag{14}$$

Then $\hat{g}(z)$ is a rational function of type $[m, m]$ and $g(z) - \hat{g}(z) = o(z^{m+1})$. Moreover, if 1 is not an eigenvalue of H_m, then $\hat{g}(z)$ is defined in a neighborhood of $z = 1$. Finally, if $\mathcal{K}_m$ is P-invariant, then $g(z) = \hat{g}(z)$.

[4] In particular, it is sufficient to take any $m \geq \nu$, where $\nu \leq N$ is the degree of the minimal polynomial of P, that is, the monic polynomial p of minimal degree such that $p(P) = 0_{N \times N}$: this is a consequence of the Cayley-Hamilton theorem.

Algorithm 1 Arnoldi based calculation of $\hat{g}(z)$

Input: $m \geq 1$; a black-box mechanism for computing the function $u \mapsto Pu$
Output: a triple (a_0, V_m, H_m)

1: $a_0 = e_N^T e_1$
2: $v_1 = \tilde{e}_1 / \|\tilde{e}_1\|_2$
3: **for** $j = 1, 2, ..., m$ **do**
4: **for** $i = 1, 2, ..., j$ **do** $h_{ij} = (Pv_j, v_i)$
5: $w_j = Pv_j - \sum_{i=1}^{j} h_{ij} v_i$
6: $h_{j+1,j} = \|w_j\|_2$
7: **if** $(h_{j+1,j} = 0) \vee (j = m)$ **then break else** $v_{j+1} = w_j / h_{j+1,j}$
8: $m = j$
9: $V_m = [v_1, ..., v_m]$
10: **return** (a_0, V_m, H_m)

Remark 1 Note from (14) that, while $\hat{g}(z)$ is a rational function of type $[m, m]$, it is not guaranteed that $g(z) - \hat{g}(z) = o(z^{2m+1})$. Thus $\hat{g}(z)$ is not, in general, a Padé approximant, but only a weaker *Padé-type* approximant.

Comparing (7) and (14), we further see that how well $\hat{g}(z)$ approximates $g(z)$ depends on how well the polynomial $\det(I_m - zH_m)$ approximates the polynomial $\det(I - zP)$. We have already noted that the roots of these polynomials are the reciprocals of nonzero eigenvalues of H_m and P, respectively. It is known for general matrices P that, already for small values of m, H_m's eigenvalues - known as *Ritz values* in the literature - tend to be excellent approximations of the eigenvalues of P that are at the extreme of the spectrum, that is, those of either large or small modulus. The details and nature of such approximation are not yet fully understood, but in the specialized literature there is abundant experimental and theoretical evidence about it, see e.g. [18, 11]. In any case, this fact retrospectively sheds light on the somewhat surprising effectiveness of Padé approximation.

Concerning the derivatives $\hat{g}^{(j)}(x) = \frac{d^j}{dx^j}\hat{g}(x)$, which can be used to approximate the moments of T and compute as in (3), we have the following result, which stems from the familiar rules for derivation.

Proposition 3 *For $1 \leq j \leq m$, we have $g^{(j)}(x) - \hat{g}^{(j)}(x) = o(x^{m+1-j})$ in a neighborhood of the origin. Moreover, assuming that $\hat{g}^{(j)}(x)$ is defined in x, it holds $\hat{g}^{(j)}(x) = j! V_m H^{j-1} (I_m - xH_m)^{-(j+1)} \tilde{e}_1$.*

We now show that the matrices V_m and H_m can be computed via an effective and numerically stable procedure known as the *Arnoldi process* [3, 18]. Arnoldi does not require knowledge of the full matrix P, but only a black-box access to the matrix-vector multiplication function $u \mapsto Pu$, which makes it compatible with an on-the-fly approach. Algorithm 1 works incrementally and, for $j = 1, ..., m$, builds an orthormal basis of $\mathcal{K}_j$, $V_j = [v_1, ..., v_j]$, and the corresponding projected version of P onto $\mathcal{K}_j$, H_j. The next vector v_{j+1} is built by orthonormalizing Pv_j against the available basis V_j (lines 4–7, which are essentially the Gram-Schmidt orthonormalization.) If this process results in the null vector ($h_{j+1,j} = 0$), then Pv_j is linearly dependent from vectors in V_j, thus the space $\mathcal{K}_j$ is P-invariant, and the main iteration stops.

The algorithm makes use of the following variables, for $j = 1, ..., m$: the scalars $a_0, a_j \in \mathbb{R}$; vectors $v_j, w_j \in \mathbb{R}^N$; the matrix $H_m \in \mathbb{R}^{m \times m}$ whose nonzero elements are the reals $h_{l,l'}$ for $1 \leq l \leq l' + 1$ and $l \leq l' \leq m$; the matrix $V_m \in \mathbb{R}^{N \times m}$. In line 4, $(\cdot, \cdot)$ denotes inner product. Of course, Pv_j needs to be computed only once per each j-iteration. Nesting of blocks is defined by indentation. The algorithm can take advantage of a sparse storage scheme. In what follows, we let W be the maximal number of nonzero elements in $P^j \tilde{e}_1$, for any $0 \leq j \leq m$, and B the maximal number of outgoing transitions from any state. Note that $W \cdot B$ is upper bounded by the overall number of transitions. Recall that a square matrix is in upper *Hessenberg* form if all its entries below the main subdiagonal are 0.

Theorem 4 *Let $m \geq 1$ and let (a_0, V_m, H_m) be the output returned by Algorithm 1. Then V_m is an orthonormal basis of $\mathcal{K}_m$ and* (13) *is satisfied. As a consequence, $\hat{g}(z)$ satisfies* (14) *given this choice of a_0, V_m, H_m. Moreover, H_m is in upper Hessenberg form and if $h_{m+1,m} = 0$ then $\hat{g}(z) = g(z)$. Assuming $u \mapsto Pu$ can be computed in $O(WB)$ operations and $O(W)$ storage, the algorithm takes $O(mWB)$ operations and $O(mW)$ storage to complete.*

A numerical example illustrating Algorithm 1 is reported in Appendix C.Note that in principle we can calculate $\hat{g}(1)$, and more generally $\hat{g}(x)$ whenever defined for x, directly using the definition (14). However, it is computationally much better to proceed as follows.

$$\hat{g}(x) = a_0 + e_N^T V_m y \quad \text{with } y \text{ the (unique) solution of the system} \tag{15}$$
$$(I_m - xH_m)y = V_m^T \tilde{e}_1 \;=\; e_1^{(m)} \cdot \|\tilde{e}_1\|_2$$

(here $e_1^{(m)}$ is the first canonical vector of $\mathbb{R}^m$.) Since $(I_m - xH_m)$ is still quasi-triangular (upper Hessenberg), the system above can be solved with $O(m^2)$ operations, while $e_N^T V_m y$ takes just $O(m)$. Similarly, the first derivative, $g^{(1)}(x)$, can be computed as $e_N^T V_m y^{(1)}$, where $y^{(1)}$ is a (unique) solution of $(I_m - xH_m)y^{(1)} = y$. For $j \geq 2$, the j-th derivative $\hat{g}^{(j)}(x)$ can be computed according to the following recursion, which can be directly derived from the formula in Proposition 3 and from the matrix identity $A(I-A)^{-1} = (I-A)^{-1}A$.

$$\hat{g}^{(j)}(x) = e_N^T V_m y^{(j)} \quad \text{with } y^{(j)} \text{ the (unique) solution of the system} \tag{16}$$
$$(I_m - xH_m)y^{(j)} = jH_m y^{(j-1)}\,.$$

Thus, $\hat{g}^{(j)}(x)$ can be computed with $O(jm^2)$ operations. In the end, via the Arnoldi algorithm 1 (cost $O(mWB)$), we have reduced the computation of all the important properties of the system to the resolution of small quasi-triangular systems, which cost approximately $O(m^2)$, for a small, "affordable" m.

Remark 2 If desired, from a_0, V_m, H_m it is also possible to recover explicitly the coefficients of the polynomials $\hat{r}(z)$ and $\hat{t}(z)$ such that $\hat{g}(z) = \hat{r}(z)/\hat{t}(z)$, at an extra cost of $O(m^2)$. The interested reader is referred to Appendix A, Remark A1.

5.2 Steady state distribution and approximation error

Just like in traditional iterative methods, a direct estimation of the approximation error $|g(z) - \hat{g}(z)|$ turns out to be not feasible. We can, however, get indications on the quality of the approximation by estimating the norm of a certain residual vector. In the rest of the section, we will assume that the chain has a limit distribution starting from the initial state vector e_1, that is, there exists

$$\pi \stackrel{\triangle}{=} \lim_{i\to\infty} P^i e_1\,.$$

This distribution is of course steady state, that is $P\pi = \pi$. Also note that $p_{reach} = \pi(N)$, the N-th component of π. In what follows we will give a method to compute an approximation $\hat{\pi}$ of π based on the algorithm in the previous subsection and a measure of its quality. It will turn out that $\hat{\pi}(N) = \hat{g}(1)$, so this way we will get also a measure of the quality of $\hat{g}(z)$ near 1.

Let V_m, H_m be the matrices returned by the Arnoldi algorithm. Let $\mathcal{K} = \text{span}\{\tilde{e}_1, P\tilde{e}_1, ...\}$ be the space reachable from $\tilde{e}_1$ (note that $\mathcal{K} = \mathcal{K}_\nu$ for suitably large integer ν.) Then $\pi \in e_1 + \mathcal{K}$: indeed, for each $j \geq 0$, $P^j e_1$ can be written as $P^j e_1 = e_1 + (Pe_1 - e_1) + (P^2 e_1 - Pe_1) + \cdots (P^j e_1 - P^{j-1}e_1) = e_1 + (\tilde{e}_1 + P\tilde{e}_1 + \cdots P^{j-1}\tilde{e}_1)$. So $P^j e_1 \in e_1 + \mathcal{K}$. Since $P^j e_1 \to \pi$ and $e_1 + \mathcal{K}$ is topologically closed, we have that $\pi \in e_1 + \mathcal{K}$ as well. In view of this fact, a strategy to approximate π is therefore to find some $\hat{\pi}$ in the smaller space $e_1 + \mathcal{K}_m \subseteq e_1 + \mathcal{K}$ that makes the norm of the residual vector

$$r \;\stackrel{\triangle}{=}\; (I - P)\hat{\pi} \tag{17}$$

as small as possible. In order to understand what $\hat{\pi}$ should be, it is convenient to define vectorial versions of g and $\hat{g}$. Note that, by (7), we can write $g(z) = e_N^T \vec{g}(z)$, where $\vec{g}(z) = e_1 + (I - zP)^{-1}\tilde{e}_1$ is a vectorial version of $g(z)$. By essentially the same proof as of Theorem 1, we can show that $\vec{g}(1) = \pi$. The corresponding vectorial approximation is, taking (14) into account, $\vec{\hat{g}}(z) = e_1 + V_m(I_m - zH_m)^{-1}V_m^T\tilde{e}_1$, and, by essentially the same proof as of Theorem 3, we have $\vec{g}(z) - \vec{\hat{g}}(z) = o(z^{m+1})$, componentwise. So it is natural to take $\hat{\pi} \stackrel{\triangle}{=} \vec{\hat{g}}(1) = e_1 + V_m(I_m - H_m)^{-1}V_m^T\tilde{e}_1$, if defined. In the end, this is equivalent to the following

$$\hat{\pi} \;\stackrel{\triangle}{=}\; e_1 + V_m y \quad \text{with } y \text{ a (unique) solution of } (I_m - H_m)y \;=\; V_m^T\tilde{e}_1\,. \tag{18}$$

If H_m has not 1 as an eigenvalue, then (18) has a unique solution[5] that can be computed in time and space $O(m^2)$. Furthermore, taking Theorem 3 into account, we have that $\hat{\pi}(N) = \hat{g}(1)$.

[5] In case 1 is an eigenvalue of H_m, the set of solutions of (18) is an affine space of dimension 1, and one might want to apply further constraints to select a solution y, like imposing that the resulting $\hat{\pi}$ is stochastic; we do not pursue this here. Also note that $V_m^T\tilde{e}_1 = e_1^{(m)} \cdot \|\tilde{e}_1\|_2$.

Finally, the residual vector (17) can be computed quite inexpensively, that is, without having to compute $(I - P)\hat{\pi}$, which makes it possible to use it in an adaptive termination condition, in Algorithm 1, line 7. This is stated in the following proposition, which is a variation of [18, Prop.6.7]. The extra dominant cost, with respect Algorithm 1, is the computation of the solution y.

Proposition 4 $\|r\|_2 = |h_{m+1,m} \cdot y(m)|$, *where* $y(m)$ *is the* m*-th component of* y.

6 Experiments

We have put an on-the-fly implementation (in Matlab) of Algorithm 1 at work on a few simple probabilistic systems. With two exceptions, the chosen systems are relatively small, but, due to the presence of nearly uncoupled strongly connected components, they exhibit the bad separation phenomenon discussed in Section 4. In each case, the reachability probability of interest is easy to compute analytically, due to the symmetry of the system. We give below an informal description of these systems.

The $Nasty(n, \delta)$ systems have been introduced in Example 1; here we have fixed $\delta = 10^{-3}$ and considered $n = 10^5, 10^6$. A $Queue(n)$ system consists of n queueing processes, each of capacity four, running in parallel. At each time, either an enqueue (prob. 0.1) or a dequeue (prob. 0.9) request arrives, and is served by any process that is able to serve it. In case of global overflow, each process chooses either of two indeterminate error states, and remains there forever. This gives rise to 2^n possible overflow configurations, which are absorbing. The event of interest is that the system eventually reaches one specific such configuration; here the cases $n = 2, 3$ are considered. An $Ising(n)$ system consists of n particles, each of which can take on, at each time, either the *up* or the *down* spin value, with a probability depending on how many *up*'s are in system and on a temperature parameter. The *all-up* and *all-down* configurations are absorbing, and the event of interest is that all-up is eventually reached, starting from the equilibrium configuration with $n/2$ particles down and $n/2$ up; here, the cases $n = 6, 8$ are considered. In $Chemical(n)$, a solution is initially composed by $n/2$ reactant pairs of type 1 and $n/2$ reactant pairs of type 2. A reactant pair of one type can get transformed into a pair of the other type, according to a chemical rule obeying the law of mass action – the probability that the reaction occurs depends on the concentration of the reactants. A solution consisting of reactants pairs all of the same type is absorbing. The probability of eventually reaching one specific such solution is seeked for; here, the cases $n = 24, 26$ are considered.

Table 1 displays the outcomes of these experiments. In all the considered cases, Algorithm 1 returned, in reasonable time, quite accurate results in terms of relative error. For comparison, results obtained with Prism, a state-of-the-art probabilistic model checker [15], are also included. Results provided by Prism were reasonably accurate only in three out of eight cases. In one case (*Nasty* with 10^6 states), Prism was not able to build the model after about one hour. In five out of eight cases, the Matlab implementation of Algorithm 1 run anyway faster than Prism. This is *not* to say, of course, that Algorithm 1 should be preferred over traditional tools. In particular, it appears that the sophisticated symbolic data structures employed in Prism and similar tools give a definite advantage over on-the-fly algorithms in the presence of symmetries, as for instance arising from interleaving and synchronization of subsystems.The Matlab implementation, as well as the Matlab and Prism specifications of the considered examples, are available at [8].

7 Conclusion, further and related work

We have demonstrated that, in the analysis of Markov chains, generating functions provide a bridge toward Padé approximation theory that is useful both at a conceptual and at a technical level.

Although it is known that a number of interesting properties can be reduced to reachability - including until properties, see e.g. [4, Ch.10] - direct extensions of the method to full temporal logics, such as LTL, seem worth studying, as well as extensions to richer models, like continuous Markov chains or Markov Decision Processes. Another potential field of application is the time-bounded analysis of infinite-state system.

As mentioned in Section 6, methods and tools based on a symbolic representations of the entire state-space, such as Prism [15], can take great advantage of the presence of system regularities: in those cases, an on-the-fly approach cannot be expected to match the performance of these tools. Nevertheless, as indicated by our

System			Algorithm 1				Prism			
Name	N	p_{reach}	$\hat{p}_{reach}$	$\%error$	m	$time$	$\hat{p}_{reach}$	$\%error$	m	$time$
Nasty	10^5	0.5	0.5000	$< 10^{-11}$	5	0.08	0.4999	$< 10^{-4}$	27	5524
	10^6	0.5	0.5000	$< 10^{-10}$	5	0.06	-	-	-	-
Queue	49	0.25	0.2500	$< 10^{-5}$	15	0.92	0.2454	1.81	133176	0.34
	231	0.125	0.1248	0.11	37	11.21	0.0013	98.91	1983765	22.63
Ising	64	0.5	0.5000	$< 10^{-3}$	9	2.06	0.4982	0.34	26433	0.05
	256	0.5	0.4998	<0.10	18	20.58	0.0578	88.42	1257073	8.16
Chemical	25	0.5	0.4994	0.10	18	0.40	1.2×10^{-7}	99.99	2000030	1.55
	27	0.5	0.4937	1.25	20	0.45	7.8×10^{-9}	99.99	2000034	1.96

Table 1: Results of some experiments. N = number of states; p_{reach} = exact reachability probability. $\hat{p}_{reach}$ = probability returned by the algorithms, truncated at the 4-th digit after decimal point; $\%error = 100 \times |p_{reach} - \hat{p}_{reach}| / p_{reach}$ is the relative error percentage; m = number of iterations; *time* = time in seconds. For Algorithm 1, formula (15) has been employed. For Prism, default options have been employed except that for m (tweaking other options did not lead to significant improvements); model building time is included in *time*. Experiments run on a Lenovo Thinkpad machine, with Intel Core i7-4500U CPU at 1.8GHz and 8GB RAM, under Windows 8.1.

small-scale experiment, the presented methodology turns out to be helpful in situations of bad eigenvalues separation, and/or when, for whatever reason, building the entire system's model turns out to be not feasible. A quite different on-the-fly approach is considered by Latella et al. [16]: what they do might be described, in the present context, as considering polynomial approximations of the form (11), with a transition matrix restricted to transient states. As we have argued, convergence of such polynomial approximations can be quite slow. The matrix $(I - zP)^{-1}$ we have considered here seems to be connected to, and somehow generalizes, the "fundamental matrix" of absorbing chains of Kemeny and Snell [14]: this aspect too deserves further investigation. Connections with formal power series and weighted automata [6] prompt further research questions.

Considering the intuitive connections that can be established, there is surprisingly little work on application of generating functions to Markov chains analysis. In fact, despite intensive Googling, the only reference to this connection we have found in the literature is a manuscript by P. Doyle [9], where this possibility is explicitly hinted at and briefly elaborated. No connection with Padé approximation theory is made, though. In numerical linear algebra, numerous works are devoted to the experimental evaluation of projective methods applied to Markov chains, see e.g. [17] and references therein. These works focus on the calculation of steady-state probabilities and no connection to generating functions is made.

On the other hand, Padé approximation and projective methods have been widely employed in Engineering, at least starting from the early 1990's, in the construction of so-called "reduced order" models. Essentially, the goal there is to build low-dimensional approximations, more amenable to analysis, of large linear dynamical systems, while preserving certain properties of interest, like stability. The formal analog there of our generating function $g(z)$ is the system's transfer function $H(z)$, obtained by applying the Laplace transform to the linear differential equations describing the system. There is a large body of literature on this subject: the interested reader might start from e.g. [10, 2], and follow the references therein.

References

[1] L. V. Ahlfors. *Complex Analysis*, 3/e. McGraw Hill Higher Education, 1980.

[2] A.C. Antoulas. *Approximation of Large-scale Dynamical Systems*. SIAM, 2005.

[3] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, vol. 9, pp. 17-29, 1951.

[4] C. Baier, J-P. Katoen. *Principles of model checking*. MIT Press, 2008.

[5] G. Baker Jr. *Essentials of Padé Approximants*. Academic Press, 1975.

[6] M. Boreale. Weighted Bisimulation in Linear Algebraic Form. *Proc. of CONCUR 2009*, LNCS 5710, pp. 163-177, Springer, 2009.

[7] M. Boreale. Analysis of probabilistic systems via generating functions and Padé approximation. *Proc. of ICALP 2015*, LNCS 9135, pp. 82–94, Springer, 2009.

[8] M. Boreale. Matlab implementation and PRISM specifications of the examples in Section 6 of the present paper. Available from `http://rap.dsi.unifi.it/~boreale/papers/GFviaKrylov.rar`

[9] P. Doyle. Markov chains via generating functions. Manuscript, 2010. `https://math.dartmouth.edu/~doyle/docs/mc/mc.ps`

[10] P. Feldmann, R.W. Freund. Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14(5), 1995.

[11] E.J. Grimme. *Krylov projection methods for model reduction*. PhD Thesis, University of Illinois at Urbana-Champaign, 1997.

[12] M.H. Gutknecht. Stable Row Recurrences for the Padé Table and Generically Superfast Look-ahead Solvers for Non-Hermitian Toeplitz Systems. *Linear Algebra Appl.* 188(89), pp. 351-421, 1993

[13] D.J. Hartfiel, Carl D. Meyer. On the structure of stochastic matrices with a subdominant eigenvalue near 1. *Linear Algebra And Its Applications*. vol. 272, pp. 193-203, 1998.

[14] J.G. Kemeny, J.L. Snell. *Finite Markov Chains*. Undergraduate Texts in Mathematics, Springer, 1983.

[15] M. Kwiatkowska, G. Norman and D. Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Proc. 23rd International Conference on Computer Aided Verification* (CAV'11), vol. 6806 of LNCS, pages 585-591, Springer, 2011.

[16] D. Latella, M. Loreti, M. Massink. On-the-fly Probabilistic Model Checking. *ICE 2014*, EPTCS 166, pp. 45-59, 2014.

[17] B. Philippe, Y. Saad , W.J. Stewart. Numerical Methods in Markov Chain Modelling. *Operations Research*, vol. 40, pp. 1156–1179, 1996.

[18] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[19] H.S. Wilf. *Generatingfunctionology*, 2/e. Academic Press, 1994. `http://www.math.upenn.edu/~wilf/gfology2.pdf`

A Proofs

Theorem A1 (Theorem 1) *There is a real $R > 1$ such that the power series $g(z)$ in (2) converges for all $|z| < R$. Moreover, there is a rational function $r(z)/t(z)$ such that for all such z's*

$$\textbf{(a)}\ \ t(z) \neq 0 \qquad \textbf{(b)}\ \ \deg(r), \deg(t) \leq N-1 \qquad \textbf{(c)}\ \ g(z) = r(z)/t(z)\,.$$

Proof Let $c_j \stackrel{\triangle}{=} e_N^T P^j e_1$ denote the probability that $X_j = N$, for $j \geq 0$. Since N is absorbing, easy calculations show that, for $j \geq 1$, $a_j = c_j - c_{j-1}$. Therefore, we have, for $j \geq 1$:

$$\begin{aligned} a_j &= c_j - c_{j-1} \\ &= e_N^T P^j e_1 - e_N^T P^{j-1} e_1 \\ &= e_N^T P^{j-1} \tilde{e}_1 \end{aligned}$$

where $\tilde{e}_1 \stackrel{\triangle}{=} (P - I)e_1$, with I the $N \times N$ identity matrix. Therefore

$$g(z) = a_0 + z \cdot \sum_{j \geq 0} e_N^T z^j P^j \tilde{e}_1\,. \tag{19}$$

Now, consider $|z| < 1$. We first note that $(I - zP)^{-1}$ exists for such z's. In fact, the roots of $\det(I - zP) = z^N \det((1/z)I - P)$, seen as a polynomial in z, are just the reciprocals of the nonzero roots of $\det(zI - P)$: the latter is the characteristic polynomial of P, whose roots are just P's eigenvalues. From Perron-Frobenius, we know that the an eigenvalue of maximal modulus of P is 1. This implies that the minimal modulus of any root of $\det(I - zP)$ is ≥ 1. Now a direct calculation shows that, again for $|z| < 1$

$$(I - zP) \cdot (P^0 + zP^1 + z^2P^2 + \cdots) = I$$

where the convergence, under any norm, on the left-hand side is guaranteed by the fact that the matrix zP has eigenvalues of modulus < 1. This implies that $(I - zP)^{-1} = (P^0 + zP^1 + z^2P^2 + \cdots)$. From this fact and (19), we have

$$\begin{aligned} g(z) &= a_0 + z \cdot \sum_{j \geq 0} e_N^T z^j P^j \tilde{e}_1 \\ &= a_0 + z \cdot e_N^T (\sum_{j \geq 0} z^j P^j) \tilde{e}_1 \\ &= a_0 + z \cdot e_N^T (I - zP)^{-1} \tilde{e}_1\,. \end{aligned} \tag{20}$$

Now, by the Cramer rule, we know that

$$(I - zP)^{-1} = \frac{\mathrm{Adj}(I - zP)}{\det(I - zP)} \tag{21}$$

where $\mathrm{Adj}(I - zP)$ is the adjoint matrix of $(I - zP)$, whose element of row i and column j is given by $(-1)^{i+j} \det((I - zP)_{ji})$, with $(I - zP)_{ji}$ the matrix obtained from $(I - zP)$ by deleting row j and column i. Moreover, seen as polynomials, $\deg(\det(I - zP)) \leq N$ while each entry of $\mathrm{Adj}(I - zP)$ has degree $\leq N - 1$. Taking (20) an (21) into account, we see that, for $|z| < 1$, $g(z)$ can be expressed as a rational function

$$g(z) = a_0 + \frac{z \cdot e_N^T \mathrm{Adj}(I - zP)\tilde{e}_1}{\det(I - zP)} = a_0 + \frac{\tilde{r}(z)}{\tilde{t}(z)} \tag{22}$$

with $\deg(\tilde{r}) \leq N$ and $\deg(\tilde{t}) \leq N$. Equation (22) also tells us that the coefficients of the Taylor expansion around 0 of $a_0 + \frac{\tilde{r}(z)}{\tilde{t}(z)}$ are exactly the moments of $g(z)$. Let us now study the poles of minimal modulus of $\tilde{r}(z)/\tilde{t}(z)$, if any. Any such pole must be a $z_0 \in \mathbb{C}$ whose multiplicity as a root of $\tilde{t}(z)$ is strictly greater than its multiplicity as root of $\tilde{r}(z)$; moreover there must be no z_0' of smaller modulus with this property. We already know that $\tilde{t}(z) = \det(I - zP)$ has no root for $|z| < 1$. Let z_0 be any complex number on the unit circle that is a root of $\det(I - zP)$; note that, from Perron-Frobenius, we know that there is at least one such root, $z_0 = 1$. From the absolute convergence criterion, we know that $g(z_0) = \sum_{j \geq 0} a_j z_0^j$ is convergent and finite. Consider now $a_0 + \tilde{r}(xz_0)/\tilde{t}(xz_0)$ as a function of the real variable x, and take its limit to 1 from the left

$$\begin{aligned} \lim_{x \to 1^-} a_0 + \frac{\tilde{r}(xz_0)}{\tilde{t}(xz_0)} &= \lim_{x \to 1^-} g(xz_0) \\ &= g(z_0) \end{aligned}$$

where, in the second equality, we have used Abel's limit theorem for power series, see e.g. [1]. Since this limit exists finite, the multiplicity of z_0 as a root of $\tilde{r}(z)$ must be at least as big as its multiplicity as a root of

$\tilde{t}(z)$, therefore z_0 is not a pole of $\tilde{r}(z)/\tilde{t}(z)$. So the smallest-modulus pole of $\tilde{r}(z)/\tilde{t}(z)$, if any, must occur at some $|z_0| > 1$. In particular, if it exists, it must be $z_0 = 1/\lambda$ for some eigenvalue λ of P such that $0 < |\lambda| < 1$. Thus the function $a_0 + \tilde{r}(z)/\tilde{t}(z)$ is defined in an open disk of radius $R > 1$ centered at the origin; in this disk, the function is analytic, that is it coincides with the power series in z of its Taylor coefficients. But this power series is exactly $g(z)$, as implied by (22), therefore $g(z) = a_0 + \tilde{r}(z)/\tilde{t}(z)$ for $|z| < R$. Via simple manipulations, a new rational expression for $g(z)$ can therefore be obtained

$$g(z) = \frac{r(z)}{t(z)}$$

where $\deg(r), \deg(t) \leq N - 1$, as resulting from the cancelation of the common root 1 of $\tilde{r}(z)$ and $\tilde{t}(z)$. □

Corollary A1 (Corollary 1) *There is $R > 1$ such that for $|z| < R$ and for $\tilde{e}_1 = (P - I)e_1$*

$$g(z) = a_0 + z \cdot e_N^T (I - zP)^{-1} \tilde{e}_1 .$$

In particular, $g(z)$ has a radius of convergence either $R = |1/\lambda|$ for some eigenvalue $0 < |\lambda| < 1$ of P, or $R = +\infty$.

Proof The statement follows from arguments in the last part of the proof of Theorem 1. □

Theorem A2 (Theorem 3) *Let $m \geq 1$. Consider the function, defined in a neighborhood of the origin*

$$\hat{g}(z) \stackrel{\triangle}{=} a_0 + z \cdot (e_N^T V_m)(I_m - zH_m)^{-1}(V_m^T \tilde{e}_1) .$$

Then $\hat{g}(z)$ is a rational function of type $[m, m]$ and $g(z) - \hat{g}(z) = o(z^{m+1})$. Moreover, if 1 is not an eigenvalue of H_m, then $\hat{g}(z)$ is defined in a neighborhood of $z = 1$. Finally, if $\mathcal{K}_m$ is P-invariant, then $g(z) = \hat{g}(z)$.

Proof We have seen (Corollary 1) that $g(z) = a_0 + z(I - zP)^{-1}\tilde{e}_1 = a_0 + z \cdot \sum_{i \geq 0} z^i e_N^T P^i \tilde{e}_1$, within an appropriate disk $|z| < R$ centered at the origin. Consider now the application ϕ defined with respect to the canonical basis in $\mathbb{R}^N$ by the matrix $V_m H_m V_m^T$: ϕ is just an orthogonal projection onto $\mathcal{K}_m$, followed by an application of P, followed by another orthogonal projection onto $\mathcal{K}_m$. For any $u \in \mathcal{K}_m$ such that $Pu \in \mathcal{K}_m$ as well, it is then easy to show that $\phi(u) = Pu$, that is $Pu = V_m H_m V_m^T u$. Using this fact and $V_m^T V_m = I_m$ (orthonormality of the chosen basis of $\mathcal{K}_m$), one shows by induction on i that, for each $0 \leq i \leq m - 1$, $P^i \tilde{e}_1 = V_m H^i V_m^T \tilde{e}_1$. Hence the moments a_j can be expressed thus for $1 \leq j \leq m$

$$a_j \quad = \quad (e_N^T V_m) H_m^j (V_m^T \tilde{e}_1) . \tag{23}$$

Now, for sufficiently small $|z| < 1$, $(I_m - zH_m)^{-1}$ exists and the expression for $\hat{g}(z)$ can therefore be expanded as the power series:

$$\hat{g}(z) = a_0 + z \sum_{j \geq 0} z^j (e_N^T V_m) H_m^j (V_m^T \tilde{e}_1) = a_0 + \sum_{1 \leq j \leq m} z^j a_j + o(z^{m+1}) .$$

This is sufficient to prove that $g(z) - \hat{g}(z) = o(z^{m+1})$. Moreover, from its expression (14), it is evident that $\hat{g}(z)$ is a rational function of type $[m, m]$. Next, note that if 1 is not an eigenvalue of H_m, then there is a neighborhood of 1 where $(I_m - zH_m)^{-1}$ exists, and therefore $\hat{g}(z)$ is defined in this neighborhood.

Finally, assume $\mathcal{K}_m$ is P-invariant: then for each $j \geq 0$ both $P^j \tilde{e}_1$ and $P^{j+1} \tilde{e}_1$ are in $\mathcal{K}_m$. As a consequence, the equality (23) can be also extended to all $j > m$. This implies that $g(z) = \hat{g}(z)$. □

Theorem A3 (Theorem 4) *Let $m \geq 1$ and let (a_0, V_m, H_m) be the output returned by Algorithm 1. Then V_m is an orthonormal basis of $\mathcal{K}_m$ and* (13) *is satisfied. As a consequence, $\hat{g}(z)$ satisfies* (14) *given this choice of a_0, V_m, H_m. Moreover, H_m is in upper Hessenberg form and if $h_{m+1,m} = 0$ then $\hat{g}(z) = g(z)$. Assuming $u \mapsto Pu$ can be computed in $O(WB)$ operations and $O(W)$ storage, the algorithm takes $O(mWB)$ operations and $O(mW)$ storage to complete.*

Proof See [18, Proposition 6.4–6.6]. The part on complexity is obvious if a suitable sparse storage scheme is adopted. □

Proposition A1 (Proposition 4) *$\|r\|_2 = |h_{m+1,m} \cdot y(m)|$, where $y(m)$ is the m-th component of y.*

Proof See the proof of [18, Prop.6.7], with $A = I - P$, $b = 0$ and $x_0 = 0$ (vectors), $\beta = \|\tilde{e}_1\|_2$. □

Remark A1 (computing $\hat{r}(z)/\hat{t}(z)$) We explain how to modify Algorithm 1 to compute polynomials $\hat{r}(z), \hat{t}(z)$ such that $\hat{g}(z) = \hat{r}(z)/\hat{t}(z)$. Let $\beta \triangleq \|\tilde{e}_1\|_2$. For $j = 1, .., m$, consider the matrix $\overline{H}_j \in \mathbb{R}^{(m+1)\times m}$, obtained by adding to H_j the vector $(0, ..., 0, h_{j+1,j})$ as a last row, and the sequence of vectors $u_j \in R^j$, defined thus

$$u_1 = \beta$$
$$u_{j+1} = \overline{H}_j u_j \, .$$

Exploiting the following relation which holds true for the Arnoldi process (see [18, Proposition 6.5])

$$PV_j = V_{j+1}\overline{H}_j \tag{24}$$

one can show by induction on j that u_j is the vector of the coordinates of $P^j\tilde{e}_1$ in the basis $V_j = [v_1, ..., v_j]$, that is $P^j\tilde{e}_1 = V_j u_j$. Therefore $a_j = e_N^T P^j \tilde{e}_1 = e_N^T V_j u_j$, for $j = 1, ..., m$. Note that computing a_j this way has an extra cost of $O(m)$. Therefore we can compute the first $m+1$ moments $a_0, a_1, ..., a_m$ without resorting to explicit power iteration and at a global extra cost of $O(m^2)$. Lines to to this effect can be inserted in the main iteration (j-loop) of Algorithm 1. Once the main loop has terminated, it is possible to get an eigendecomposition - eigenvalues together with their algebraic multiplicities - of H_m via standard methods, like the QR decomposition, which costs $O(m^2)$, given the Hessenberg form of H_m. From this information it is possible to reconstruct $\tilde{t}(z) = \det(I - zP)$ with a cost of $O(m^2)$. Now we know there is $\tilde{r}(z)$ of degree $\leq m$ such that $\hat{g}(z) = a_0 + \tilde{r}(z)/\tilde{t}(z)$. Clearly, one must have $\tilde{r}(z)/\tilde{t}(z) = (\sum_{j=1}^{m} a_j z^j) + o(z^{m+1})$, which implies $\tilde{r}(z) = (\sum_{j=1}^{m} a_j z^j)\tilde{t}(z) \bmod z^{m+1}$. Since $\deg(r) \leq m$, the last equality in fact defines $\tilde{r}(z)$ via $\tilde{r}(z) = (\sum_{j=1}^{m} a_j z^j)\tilde{t}(z)$. This last step costs another $O(m^2)$. In the end, $\hat{g}(z) = a_0 + \tilde{r}(z)/\tilde{t}(z)$, from which $\hat{r}(z), \hat{t}(z)$ can be derived explicitly if desired.

B Bad separation of eigenvalues

It is well-known that that the bad separation phenomenon - subdominant eigenvalues close to 1 - occurs if there is a proper subset $\mathcal{S}_0$ of the chain's states that are strongly coupled with one another, but nearly uncoupled with other states in the chain. More precisely, assuming - possibly after a rearrangement - $\mathcal{S}_0$ consists of consecutive integers, the diagonal block in P corresponding to $\mathcal{S}_0$, say P_0, forms a nearly stochastic matrix. By continuity arguments, the spectral radius of P_0 will be close to 1, which will force P to have a close to 1 eigenvalue in turn. This is more clearly seen in the special case $\mathcal{S}_0$ is a whole (non terminal) strongly connected component of the chain: in that case, $\det(zI_k - P_0)$ ($k = |\mathcal{S}_0|$) is a factor of the characteristic polynomial of P, hence eigenvalues of P_0 are also eigenvalues of P. For example, in $Nasty(6, \delta)$, take $\mathcal{S}_0 = \{1, 2, 3, 4\}$: the largest eigenvalue of P_0 is $\lambda \approx 0.9993$, and this is also the second largest (subdominant) eigenvalue of P in modulus. An in-depth discussion of the relation between the structure of stochastic matrices and subdominant eigenvalues can be found in [17, 13].

We finally note that the situation of subdominant eigenvalues of P close to 1 is also one where traditional iterative methods like Jacobi and Gauss-Seidel, employed in state-of-the-art probabilistic model checkers, show very slow convergence, if not stagnation at all. We refer the reader to the literature on numerical Linear Algebra, such as [18], for a discussion of this aspect.

C Numerical examples

Consider the chain $Nasty(6, \delta)$ of Example 1. We fix $\delta = 1/1000$. Since $N = 6$, we know that $g(z)$ coincides with its [5, 5]-Padé approximant. We compute the coefficients a_i for i from 0 through 10:

$$a_0 = 0\,, \quad a_1 = \frac{1}{3000}\,, \quad a_2 = \frac{1199}{3600000}\,, \quad \cdots \quad, \quad a_{10} = \frac{6259830773409424147463909122781 9}{1.889568\times 10^{39}}\,.$$

We form the 5×5 matrix C as specified in Section 3, then solve the system $C\beta = \gamma$ for β, with $\gamma = (-a_6, ..., -a_{10})^T$, and then compute $\alpha = \tilde{C}\beta'$, thus obtaining the vectors of coefficients for $r(z)$ and $t(z)$, respectively as

$$\alpha = (\alpha_0, \alpha_1, ..., \alpha_5) = 10^{-3} \times (0, 1, \tfrac{1}{18000}, \tfrac{1}{162000000}, \tfrac{1}{1296000000000}, 0)^T$$
$$\beta' = (1, \beta_1, ..., \beta_5) = 10^{-3} \times (1000, -999, -\tfrac{1999}{6000}, -\tfrac{2999}{54000000}, -\tfrac{1333}{216000000000}, 0)^T\,.$$

Simplifying some common integer factors, we get the following form

$$g(z) = -\frac{1}{6}\,\frac{z^4 + 8000z^3 + 72000000z^2 + 432000000000z}{1333z^4 + 11996000z^3 + 71964000000z^2 + 215784000000000z - 216000000000000}\,.$$

We get again $p_{reach} = g(1) = 1/2$, and $E[T|Reach] = g(1)^{-1} \cdot g'(1) \approx 1500.25$.

Consider now the Markov chain $Nasty(N, \delta)$, this time with $N = 10^4$ and $\delta = 10^{-3}$. We apply Algorithm 1 with $m = 3$ and obtain the following (all values truncated at the fifth digit after decimal point)

$$H_m \;=\; \begin{bmatrix} 0.66581 & 0.94225 & 0.51662 \\ 0.23574 & 0.33318 & 0.18258 \\ 0 & 0.00022 & 0.00002 \end{bmatrix} \qquad \hat{g}(z) \;=\; \frac{10^{-3}\cdot(10^{-9}\cdot 4.93862z^3 + 10^{-5}\cdot 4.81525z^2 + 0.33333z)}{-10^{-8}\cdot 4.44337z^3 - 10^{-4}\cdot 3.10979z^2 - 0.99902z + 1}\,.$$

When evaluating at $z = 1$, we get $\hat{g}(1) = 0.49999\cdots$, which is very close to the true value $p_{reach} = g(1) = 0.5$. The value of $\hat{g}(1)$ computed via (15) is the same, up to the 13-th digit after decimal point. The residue norm is $\|r\|_2 = 1.01415 \times 10^{-8}$.