Topics in Markov Chain Theory and Simulation Optimisation

Berkhout, J.

2016

document version
Publisher's PDF, also known as Version of record

5. THE JUMP START POWER METHOD

This chapter is based on [32, 35].

This chapter presents a new numerical method for approximately computing the ergodic projector of a finite aperiodic Markov multi-chain. Our approach requires neither structural information on the chain, such as, the identification of ergodic classes, transient states, or qualitative information, such as, whether the chain is nearly decomposable or not. The theoretical deduction of the new method is corroborated by an extensive numerical study.

The chapter is organized as follows. In Section 5.1 an introduction into the research topic is given. Section 5.2 formally introduces Markov multi-chains, nearly decomposable Markov chains (which will be used as benchmark problems), and defines concepts used throughout the chapter. Section 5.3 presents the main technical results of the chapter. The technical results are elaborated into a numerical framework called the jump start power method (JSPM) in Section 5.4, where also a numerical study on the performance of the algorithm is conducted. This chapter concludes with a discussion of potential further research in Section 5.5.

5.1 Introduction

This chapter studies aperiodic Markov multi-chains defined on discrete state space $\mathbb{S} = \{1, \ldots, S\}$, with $S \in \mathbb{N}$. The steady-state behaviour of an aperiodic Markov multi-chain P is characterized through its ergodic projector Π_P, where

$$\Pi_P = \lim_{n\to\infty} P^n,$$

see [115, 161][1]. Computing Π_P in the above way, i.e., by taking powers of P is known as the *power method* (PM). Throughout this chapter, $\|\cdot\|$ will denote the maximum absolute row sum. Any finite aperiodic Markov chain P is *geometrically ergodic*, i.e., there exist finite numbers $r, c = \sup_{l=0,1,\ldots,r-1} \|P^l - \Pi_P\|$ and $\beta \in (0,1)$ such that

$$\forall n \geq r: \quad \|P^n - \Pi_P\| \leq c\beta^n,$$

where β is called the *rate* and r is called the *transient phase*, see, for example, [93]. For ease of reference we call r, c and β *ergodicity parameters* of P. Geometric ergodicity implies that PM enjoys a geometric rate of convergence once the powers exceed r. The main advantages of the power method is that it is easy to use and it requires no further

[1] Note that aperiodicity is not a restrictive assumption as, at the expense of little convergence speed loss, one can always mix an aperiodic P with an identity matrix to construct an aperiodic Markov chain with the same ergodic projector.

information on P. This robustness is the main reason why PM is used for the acclaimed Google PageRank algorithm introduced by [44], and for more detail see [27, 131, 70].

Iterative methods, such as PM, converge slowly in case the modulus of a subdominant eigenvalue of P is close to 1, see [105, 89]. This typically happens if either P has relatively many transient states or if P is nearly decomposable. Roughly speaking, an irreducible chain P is called nearly decomposable if the state-space can be divided into classes of states so that the interactions between states inside classes are relatively frequent compared to interactions between the classes (a formal definition will be provided later in the text). It can be shown that an irreducible Markov chain without transient states is nearly decomposable if and only if the subdominant eigenvalue is close to 1, see [83]. A famous example of a nearly decomposable Markov chain is the so-called Courtois matrix, which is a 6×6 transition matrix for which PM requires $n \approx 69.000$ in order to provide an approximation of Π_P that is correct in first 6 digits, see Chapter 3 from [173].

In case the ergodic classes and the transient states are known, one may compute the ergodic projector directly by first computing the equilibrium distribution for each ergodic class, and then the long-term behaviour of the transient states, see [115, 31] and the detailed discussion in Section 5.2. For a comprehensive overview on numerical methods for computing the ergodic projector of a finite Markov chain we refer to [173].

Our research on Markov chains is stimulated by the growing interest in the analysis of social networks (where the Markov chain is used to model relationships among social agents, see, e.g., [63, 157]) and by the analysis of the world wide web, where based on the (bored-) random-surfer-concept, the Markov chain models the probability of randomly going from one page to another, [131, 70]. A key feature of these networks is that they are large and that neither their structure (transient states, ergodic classes) nor their balancedness (nearly decomposable or not) are known *a priori*. Note that Markov chain modelled social networks are often nearly decomposable due to the tendency for people to associate with people like themselves, a phenomenon called *homophily* in social network theory, [141]. Other examples of these type of complex networks include telecommunication networks, cognitive and semantic networks and biological networks.

In this chapter a novel approach is developed for computing the ergodic projector of an aperiodic Markov chain. We firstly establish a new representation of the ergodic projector of P through constructing a alternative Markov chain and then make this result useful for numerical computation. In the rest of the chapter, let I denote an appropriate sized identity matrix. Starting point of our analysis is the known analytical relation

$$\Pi_P = \lim_{\alpha \downarrow 0} \alpha (I - (1-\alpha)P)^{-1}, \tag{5.1}$$

see, for example, Theorem 1.5 in [111], where the term $\alpha(I-(1-\alpha)P)^{-1}$ is recognizable the *resolvent kernel of* P. See also [124] for applications of the resolvent kernel in stability theory of Markov chains. We call the transformation

$$H_\alpha(P) = \alpha P (I - (1-\alpha)P)^{-1},$$

for $\alpha \in (0, 1]$, the *modified resolvent kernel of* P. Letting X_α be a geometrically distributed random variable with parameter $\alpha \in (0, 1)$, the modified resolvent kernel can be written as

$$H_\alpha(P) = \mathbb{E}[P^{X_\alpha + 1}],$$

since $\|(1-\alpha)P\| < 1$. This shows that $H_\alpha(P)$ is again a Markov transition matrix with the same ergodic projector as P for any $\alpha \in (0,1)$, and since at $\alpha = 1$ it holds $H_\alpha(P) = P$, the statement holds for $\alpha = 1$ as well. In formula,

$$\Pi_P = \Pi_{H_\alpha(P)} = \lim_{k\to\infty} (H_\alpha(P))^k, \quad \alpha \in (0,1]. \tag{5.2}$$

We will show that

$$\|\Pi_P - (H_\alpha(P))^k\| \leq (\alpha\gamma(P))^k,$$

where $||\cdot||$ denotes the maximal absolute row-sum norm, and $\gamma(P)$ a finite (possibly large) constant depending on P, to be defined later in the text.

Letting $\alpha < 1/\gamma(P)$, the result put forward in (5.1) implies that the Markov kernel $H_\alpha(P)$ is geometrically ergodic with transient phase $r = 1$ and rate $\alpha\gamma(P)$. Put differently, the transformation $P \mapsto H_\alpha(P)$ provides a jump start for PM as the desired contraction property is immediately effective. Moreover, we will show that iterating the transformation yields a geometric reduction in the geometric rate, so that, for example, $H_\alpha(H_\alpha(P))$ has a rate that is proportional to α^2. The above theoretical results lead to a new numerical approach for approximately computing Π_P, called *jump start power method.*

The main contributions of this chapter are as follows.

- A new formula for the stationary projector of an aperiodic finite Markov chain is established[2].
- The error of approximating Π_P by powers of the modified resolvent $(H_\alpha(P))^k$ is of order $(\alpha\gamma(P))^k$. We use this fact to introduce the jump start power method (JSPM) that enjoys the robustness of PM but overcomes the numerical deficiency of PM. JSPM works well for multi-chains, nearly decomposable chains, and chains with a relatively large number of transient states.
- An extensive numerical study is provided that corroborates the form of the analytically bound for decay of the error and illustrates the numerical advantages of JSPM.

5.2 A Brief Review of Markov Multi-Chains

Let $\{X_t\}_{t=0,1,\ldots}$ denote a homogeneous aperiodic discrete-time Markov chain with transition matrix P defined on state space $\mathbb{S} = \{1,\ldots,S\}$, where S denotes the number of states; see [115] for definitions. For the (i,j)-th element of P it holds that $P(i,j) = \Pr(X_{t+1} = j | X_t = i)$ is independent of t and the past states, i.e., the probability distribution of the next state only depends on the current state. This leads to

$$(P^n)(i,j) = \Pr(X_n = j | X_0 = i), \qquad \text{for all } (i,j) \in \mathbb{S}\times\mathbb{S},$$

which reads as the n-step transition probability of the Markov chain, where transition matrix P^n is simply obtained from taking the n-th matrix power of P. Taking n to

[2] It is worth mentioning that in all our numerical experiments throughout this research the formula also directly applied to periodic Markov chains but we weren't able to prove this yet or to find counterexamples.

infinity leads to the (i,j)-th element of the *ergodic projector*, denoted by Π_P, and defined by

$$\Pi_P(i,j) = \lim_{n\to\infty}(P^n)(i,j), \qquad \text{for all } (i,j) \in \mathbb{S}\times\mathbb{S}.$$

Entry $\Pi_P(i,j)$ represents the probability of the chain being in state j in the long-run when starting in state i. For more details we refer to [115].

In case the Markov chain has only one closed irreducible set of states, also called *ergodic class*, and a (possible empty) set of transient states, it is called a Markov uni-chain (in short: uni-chain). For uni-chains it holds that the chain will eventually be trapped in the (unique) ergodic class, independent of the initial state. The unique distribution to which a uni-chain converges is described by the stationary distribution of P denoted as $\pi_P^\top$ which can be found by solving $\pi_P^\top P = \pi_P^\top$. Since the stationary distribution is independent of the initial state, all rows of Π_P equal $\pi_P^\top$ in case P describes Markov uni-chain.

Markov multi-chains (in short: multi-chains) have multiple ergodic classes and a (possibly empty) set of transient states. Other than for uni-chains, in case of multi-chains the initial state has impact on the resulting stationary distribution, which stems from the fact that once the chain enters one of the several ergodic classes it remains there forever. First of all one has to uncover the ergodic classes and the transient states using, for example, the already mentioned algorithm in [69]. After possible relabelling of states, the transition matrix and the ergodic projector of a multi-chain can be written in the following canonical forms, respectively,

$$P = \begin{bmatrix} P_1 & 0 & 0 & \cdots & 0 \\ 0 & P_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & P_E & 0 \\ P_{T1} & P_{T2} & \cdots & P_{TE} & P_{TT} \end{bmatrix} \quad \text{and} \quad \Pi_P = \begin{bmatrix} \Pi_1 & 0 & 0 & \cdots & 0 \\ 0 & \Pi_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \Pi_E & 0 \\ R_1 & R_2 & \cdots & R_E & 0 \end{bmatrix}, \tag{5.3}$$

where E is the number of ergodic classes. For the i-th ergodic class, P_i gives the one step transition probabilities between ergodic states from the i-th ergodic class and Π_i gives a square matrix of which all rows equal the unique stationary distribution of the chain inside the i-th ergodic class. I.e., all rows in Π_{P_i} equal $\pi_{P_i}^\top$ which is the unique probability vector satisfying $\pi_{P_i}^\top P_i = \pi_{P_i}^\top$. Note that all diagonal values of Π_P corresponding to ergodic states are non-zero contrary to the diagonal values of transient states which are zero, an insight that will be elaborated in Section 5.4.3. Hence, whether state i is ergodic or transient can be concluded from the value of entry (i,i) of Π_P. We call this criterion for ergodicity of a state the *diagonal criterion*.

Moreover, $R_i(j,k)$ gives the equilibrium probability of ending in ergodic state k (which is part of the i-th ergodic class) when starting in transient state j. In order to calculate R_i, define $\mathcal{T}$ as the number of transient states, $I_\mathcal{T}$ as the unity matrix of size $\mathcal{T}$ and $Z(j,i)$ as the probability of ending in the i-th ergodic class when starting in transient state j. Note that Z is a $\mathcal{T}\times E$ matrix Z. From, e.g., [31], it follows that

$$Z = (I_T - P_{TT})^{-1}\left[P_{T1}e_1 \quad P_{T2}e_2 \quad \cdots \quad P_{TE}e_E\right],$$

where e_i is a column vector of ones of size equal to the number of states in ergodic class i. Denote the i-th column of Z with $Z(\cdot,i)$, then it holds that $R_i = Z(\cdot,i)\pi_{P_i}^\top$.

In case there are multiple ergodic classes the stationary distribution fails to be unique. Indeed, any row of Π_P is a stationary distribution of the Markov chain. More specifically, denote the i-th row of Π_P by $\Pi_P(i,\cdot)$, then it holds that $\Pi_P(i,\cdot)$ is a probability distribution which satisfies $\Pi_P(i,\cdot)P = \Pi_P(i,\cdot)$. This implies that any convex combination of the rows is also a stationary distribution of the Markov chain, i.e., for $(\gamma_i)_{i\in\mathbb{S}} : \sum_{i=1}^{S}\gamma_i = 1$ and $\gamma_i \geq 0$, for all $i \in \mathbb{S}$, it holds that $\sum_{i=1}^{S}\gamma_i\Pi_P(i,\cdot)$ is a probability distribution which is invariant with respect to P. When an initial distribution $\mu^\top$ is considered, this convex combination is fixed (and given by $\mu^\top$) meaning that there exists an unique stationary distribution for the chain started in $\mu^\top$ (describing the long-run behaviour of the chain started in $\mu^\top$), or, more formally, $\mu^\top\Pi_P$ is the unique stationary distribution satisfying $(\mu^\top\Pi_P)P = (\mu^\top\Pi_P)$ when starting according to distribution $\mu^\top$. Literature considering Markov multi-chains includes Markov decision processes from [161], series expansion of Markov chains [19, 31, 32] and singular perturbation analysis [11, 84] where the underlying multi-chain structure is often known beforehand.

A Markov chain P is called *nearly decomposable*[3] if P is irreducible and, after possible relabelling of states, can be written in the form

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1k} \\ P_{21} & P_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & P_{(k-1)k} \\ P_{k1} & \cdots & P_{k(k-1)} & P_{kk} \end{bmatrix},$$

where the diagonal blocks P_{ii}, $i = 1, 2, \ldots, k$, are square and have rows that sum up to $1-\epsilon$, with $\epsilon > 0$ small, see, for example, [173, 148]. The multi-chain and nearly decomposability characteristic are not mutually exclusive.

Indeed, a Markov chain may be a multi-chain with transient states and may have an ergodic class that for itself constitutes a nearly decomposable chain. Below we illustrate this be means of a simple Markov chain.

Example 5.1. Let $p_1, p_2, r_1, r_2, r_3 \in (0,1)$ and define transition matrix P on state-space $\{1,2,3,4\}$ as

$$P = \begin{pmatrix} 1-p_1 & p_1 & 0 & 0 \\ p_2 & 1-p_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ r_1 & r_2 & r_3 & 1-\sum_{i=1}^{3} r_i \end{pmatrix},$$

where $0 < \sum_{i=1}^{3} r_i \leq 1$.

Markov chain P is noticeably a multi-chain with ergodic classes $\{1,2\}$ and $\{3\}$. State 4 is transient. If p_1, p_2 are small, then the submatrix describing the transitions within the ergodic class $\{1,2\}$ becomes nearly decomposable. Similar when $\sum_{i=1}^{3} r_i$ is small, state 4 becomes nearly decomposable with respect to states $\{1,2,3\}$.

[3] Some authors use the terminology *nearly completely decomposable* or *nearly completely reducible*.

The ergodic projector of P can be computed to be

$$\Pi_P = \begin{bmatrix} \frac{p_2}{p_1+p_2} & \frac{p_1}{p_1+p_2} & 0 & 0 \\ \frac{p_2}{p_1+p_2} & \frac{p_1}{p_1+p_2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{(r_1+r_2)p_2}{\left(\sum_{i=1}^3 r_i\right)(p_1+p_2)} & \frac{(r_1+r_2)p_1}{\left(\sum_{i=1}^3 r_i\right)(p_1+p_2)} & \frac{r_3}{\sum_{i=1}^3 r_i} & 0 \end{bmatrix}.$$

Note that when $p_1 = p_2 = \sum_{i=1}^3 r_i = 0$, we obtain $P = I$ and $\Pi_P = I$. ◁

In this chapter we will also study the impact of starting with a power P^q for the evaluation of Π_P and we introduce the following additional ergodicity parameters

$$\gamma(P,q) = \kappa(P,q)\phi(R,q) + \frac{c\beta^{q(\phi(P,q)+1)}}{1-\beta^q}, \tag{5.4}$$

where

$$\kappa(P,q) = \sup_{n=q,q+1,\ldots,q\phi(P,q)} \|P^n - \Pi_P\|,$$

and

$$\phi(P,q) = \left\lceil \frac{\max\{r-q,0\}}{q} \right\rceil.$$

For simplicity, $\gamma(P,1)$ is also denoted as $\gamma(P)$.

5.3 A New Formula For The Ergodic Projector

Starting point of our analysis is the equality

$$\Pi_P = (1-\alpha)\Pi_P + \alpha\Pi_P, \qquad \text{for } \alpha \in [0,1].$$

Adding and subtracting αP^q to the right hand side gives

$$\Pi_P = \alpha P^q + (1-\alpha)\Pi_P P^q + \alpha(\Pi_P - P^q),$$

where we used for the second term on the right hand side that $\Pi_P P = \Pi_P$. Inserting N times this last expression for Π_P into the first Π_P on the right hand side leads to

$$\Pi_P = \alpha P^q \sum_{n=0}^{N}((1-\alpha)P^q)^n + \Pi_P((1-\alpha)P^q)^{N+1} + \alpha(\Pi_P - P^q)\sum_{n=0}^{N}((1-\alpha)P^q)^n. \tag{5.5}$$

For $N, q \in \mathbb{N}$ and $\alpha \in (0,1)$ let

$$G_\alpha(N,P^q) = \frac{\sum_{n=0}^{N}((1-\alpha)P^q)^n}{1-(1-\alpha)^{N+1}}.$$

Note that for $\alpha \in (0,1)$ and $q \geq 1$ it holds that

$$G_\alpha(P^q) := \lim_{N\to\infty} G_\alpha(N,P^q) = \sum_{n=0}^{\infty}((1-\alpha)P^q)^n = (I-(1-\alpha)P^q)^{-1},$$

where existence of the Neumann series is guaranteed since $\|(1-\alpha)P^q\| < 1$, for $\alpha \in (0,1)$. Equation (5.5) can be rewritten in succession via (i) simplifying the second term on the right hand side, (ii) bringing the second term of the right hand side to the other side, (iii) dividing by $1-(1-\alpha)^{N+1}$ and (iv) using the $G_\alpha(N, P^q)$-notation:

$$\Pi_P = \alpha P^q G_\alpha(N, P^q) + \alpha(\Pi_P - P^q) G_\alpha(N, P^q). \tag{5.6}$$

Remark 5.1. *The (i,j)-th element of $G_\alpha(N, P^q)$ gives the scaled $(1-\alpha)$-discounted expected number of visits of the Markov chain with transition matrix P^q to the j-th state in the first $N+1$ number of discrete time steps (including the state i at time zero) when starting in state i. Intuitively, the discounting ensures that the weights of the visits after many discrete time steps of the Markov chain with transition matrix P^q are decreasing, ensuring existence of $G_\alpha(P^q)$ since $\|(1-\alpha)P^q\| < 1$, for $\alpha \in (0,1)$.*

Post-multiplying Equation (5.6) with

$$\left(\alpha P^q G_\alpha(N, P^q)\right)^{k-1}, \quad \text{for } k \in \mathbb{N}$$

i.e., the $k-1$ power of the first term of the right hand side of (5.6), gives

$$\Pi_P = \left(\alpha P^q G_\alpha(N, P^q)\right)^k + \alpha^k (\Pi_P - P^{qk}) \left(G_\alpha(N, P^q)\right)^k, \tag{5.7}$$

where we used that

$$\left(\alpha P^q G_\alpha(N, P^q)\right)^{k-1} \Pi_P = \left(\frac{\alpha \sum_{n=0}^{N} (1-\alpha)^n}{1-(1-\alpha)^{N+1}}\right)^{k-1} \Pi_P = \Pi_P.$$

Taking the limit $N \to \infty$ in (5.7) leads to

$$\Pi_P = \left(H_\alpha(P^q)\right)^k + \alpha^k (\Pi_P - P^{qk}) \left(G_\alpha(P^q)\right)^k,$$

where we use the notation

$$H_\alpha(P^q) := \alpha P^q G_\alpha(P^q),$$

which is the modified resolvent kernel of P. Analogous to $G_\alpha(\cdot)$ let in the following

$$H_\alpha(N, P^q) := \alpha P^q G_\alpha(N, P).$$

Lemma 5.1. *For $k, q \in \mathbb{N}$, $N \geq \phi(P, q)$, and $\alpha \in (0,1)$ it holds that*

$$\left\| \Pi_P - \left(H_\alpha(N, P^q)\right)^k \right\| \leq \left(\frac{\alpha \gamma(P, q)}{1-(1-\alpha)^{N+1}}\right)^k.$$

Proof. From (5.7) it follows that

$$\Pi_P - \left(H_\alpha(N, P^q)\right)^k = \alpha^k (\Pi_P - P^q) P^{q(k-1)} \left(G_\alpha(N, P^q)\right)^k. \tag{5.8}$$

Since $(\Pi_P - P^q)\Pi_P = 0$, it holds that

$$(\Pi_P - P^q) P^{q(k-1)} = (-1)^{k-1} (\Pi_P - P^q)^k,$$

so that (5.8) can be written as

$$\begin{aligned}\Pi_P - (H_\alpha(N,P^q))^k &= \alpha^k(-1)^{k-1}(\Pi_P - P^q)^k\,(G_\alpha(N,P^q)))^k \\ &= -\left[-\alpha(\Pi_P - P^q)\frac{\sum_{n=0}^{N}((1-\alpha)P^q)^n}{1-(1-\alpha)^{N+1}}\right]^k, \end{aligned} \tag{5.9}$$

where the definition of $G_\alpha(N,P^q)$ is filled in in the last equation. Applying norms to Equation (5.9) we get

$$\begin{aligned}&\|\Pi_P - (H_\alpha(N,P^q))^k\| \\ &\le \left\|\alpha(\Pi_P - P^q)\frac{\sum_{n=0}^{N}((1-\alpha)P^q)^n}{1-(1-\alpha)^{N+1}}\right\|^k \\ &\le \left(\frac{\alpha\sum_{n=0}^{N}(1-\alpha)^n\left\|\Pi_P - P^{q(n+1)}\right\|}{1-(1-\alpha)^{N+1}}\right)^k \\ &= \left(\frac{1}{1-(1-\alpha)^{N+1}}\right)^k\left\{\alpha\sum_{n_1=0}^{\min\{N,\phi(P,q)-1\}}(1-\alpha)^{n_1}\left\|\Pi_P - P^{q(n_1+1)}\right\|\right. \\ &\qquad\left.+\alpha\sum_{n_2=\phi(P,q)}^{N}(1-\alpha)^{n_2}\left\|\Pi_P - P^{q(n_2+1)}\right\|\right\}^k, \end{aligned} \tag{5.10}$$

where the summation is split at $\phi(P,q)$ into two summations, one where geometric ergodicity does not apply and one where it does, respectively. Continuing calculations from (5.10) shows

$$\begin{aligned}&\|\Pi_P - (H_\alpha(N,P^q))^k\| \\ &\le \left(\frac{1}{1-(1-\alpha)^{N+1}}\right)^k\left\{\sup_{n=0,1,\ldots,\min\{N,\phi(P,q)-1\}}\left\|\Pi_P - P^{q(n+1)}\right\|\left[1-(1-\alpha)^{\min\{N+1,\phi(P,q)\}}\right]\right. \\ &\qquad\left.+\alpha c\beta^q((1-\alpha)\beta^q)^{\phi(P,q)}\frac{1-((1-\alpha)\beta^q)^{\max\{N-\phi(P,q)+1,0\}}}{1-(1-\alpha)\beta^q}\right\}^k. \end{aligned} \tag{5.11}$$

So we may conclude from (5.11) that for

1. $N \le \phi(P,q)-1$ (geometric ergodicity does not apply):

$$\|\Pi_P - (H_\alpha(N,P^q))^k\| \le \left(\sup_{n=0,1,\ldots,N}\left\|\Pi_P - P^{q(n+1)}\right\|\right)^k$$

2. $N \ge \phi(P,q)$ (geometric ergodicity applies):
since

$$1-(1-\alpha)^{\phi(P,q)} \le \alpha\phi(P,q), \quad \text{for } \alpha\in(0,1) \text{ and } \phi(P,q) = 0,1,\ldots,$$

it holds that

$$\|\Pi_P - (H_\alpha(N,P^q))^k\| \le \left(\frac{\alpha\gamma(P,q)}{1-(1-\alpha)^{N+1}}\right)^k,$$

where $\gamma(P,q)$ is a finite constant defined in (5.4).

Note that it is necessary for the bound to be meaningful that $N \geq \phi(P,q)$ so that the geometric ergodicity applies. □

Remark 5.2. *One may ask what the best value for $\alpha \in (0,1)$ is that minimizes the norm bound from Lemma 5.1 in case of finite N. To that end, for notational easiness define the bound found in Lemma 5.1 in case $N \geq \phi(P,q)$ as $f(\alpha) = \left(\frac{\alpha\gamma(P,q)}{1-(1-\alpha)^{N+1}}\right)^k$. It holds that*

$$\lim_{\alpha\downarrow 0} f(\alpha) = \left(\frac{\gamma(P,q)}{N+1}\right)^k,$$

and

$$\lim_{\alpha\uparrow 1} f(\alpha) = (\gamma(P,q))^k,$$

so that $\lim_{\alpha\downarrow 0} f(\alpha) < \lim_{\alpha\uparrow 1} f(\alpha)$ for $k,q \in \mathbb{N}$ and $N \geq \phi(P,q)$. Furthermore, since[4]

$$\frac{d}{d\alpha} f(\alpha) = k\alpha^{k-1}\gamma(q)^k \frac{1-(1+\alpha N)(1-\alpha)^N}{[1-(1-\alpha)^{N+1}]^{k+1}} > 0, \quad \text{for } \alpha \in (0,1),$$

it holds that for any choice of k, q and $N \geq \phi(P,q)$ it is best to choose $\alpha \in (0,1)$ as small as possible.

The following theorem summarizes some properties of $H_\alpha(P^q) = \lim_{N\to\infty} H_\alpha(N, P^q)$.

Theorem 5.1. *It holds for $k,q \geq 1$ and $\alpha \in (0,1)$ that*

$$\left\|\Pi_P - (H_\alpha(P^q))^k\right\| \leq (\alpha\gamma(P,q))^k \tag{5.12}$$

and

$$\Pi_P = \lim_{\alpha\downarrow 0}(H_\alpha(P^q))^k = \lim_{q\to\infty}(H_\alpha(P^q))^k = \lim_{k\to\infty}(H_\alpha(P^q))^k. \tag{5.13}$$

Proof. Inequality (5.12) follows directly from Lemma 5.1 by letting $N \to \infty$. The first two equalities from (5.13) follow from Inequality (5.12) (note that $\lim_{q\to\infty}\gamma(P,q) = 0$) and the third equality from (5.2). □

The result put forward in Theorem 5.1 shows that for $\alpha < \gamma(P,q)$ it holds that the modified resolvent $H_\alpha(P^q)$ is geometrically ergodic with rate $\alpha\gamma(P,q)$, transient phase $r = 1$, and ergodic projector Π_P.

As our numerical study in the second part of this chapter shows, the modified resolvent is potentially more efficient than PM, which makes it, apart from the fact that it directly applies to multi-chains, an attractive alternative to PM. In the following $H_\alpha(P)$ is illustrated for Example 5.1.

Example 5.2. We revisit Example 5.1 where we assume that p_1, p_2, r_1, r_2 and r_3 are non-zero probabilities. For this chain $H_\alpha(P)$ can be explicitly solved to be

$$H_\alpha(P) = \begin{bmatrix} \frac{\alpha(1-p_1)+(1-\alpha)p_2}{\alpha+(1-\alpha)(p_1+p_2)} & \frac{p_1}{\alpha+(1-\alpha)(p_1+p_2)} & 0 & 0 \\ \frac{p_2}{\alpha+(1-\alpha)(p_1+p_2)} & \frac{\alpha(1-p_2)+(1-\alpha)p_1}{\alpha+(1-\alpha)(p_1+p_2)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{\alpha r_1+(1-\alpha)p_2(r_1+r_2)}{(\alpha+(1-\alpha)(p_1+p_2))\psi(\alpha)} & \frac{\alpha r_2+(1-\alpha)p_1(r_1+r_2)}{(\alpha+(1-\alpha)(p_1+p_2))\psi(\alpha)} & \frac{r_3}{\psi(\alpha)} & \frac{\alpha\left(1-\sum_{i=1}^3 r_i\right)}{\psi(\alpha)} \end{bmatrix},$$

[4] Note that $(1+\alpha N)(1-\alpha)^N \leq (1+\alpha)^N(1-\alpha)^N = (1-\alpha^2)^N < 1$ for $\alpha \in (0,1)$ and $N \in \mathbb{N}$.

where for notational easiness $\psi(\alpha) := \alpha + (1-\alpha)\left(\sum_{i=1}^{3} r_i\right)$. Hence, letting α tend to zero yields element-wise convergence of $H_\alpha(P)$ to Π_P, which is in accordance with Theorem 5.1. For example, the absolute error of the $(1,1)$-th element equals

$$\frac{\alpha p_1 |p_1 + p_2 - 1|}{(p_1+p_2)(\alpha + (1-\alpha)(p_1+p_2))}$$

so that the corresponding relative error is

$$\frac{|H_\alpha(P)(1,1) - \Pi_P(1,1)|}{\Pi_P(1,1)} = \frac{\alpha p_1 |p_1 + p_2 - 1|}{p_2(\alpha + (1-\alpha)(p_1+p_2))},$$

where $H_\alpha(P)(i,j)$ indicates the (i,j)-th element of $H_\alpha(P)$. It shows that the relative error of $H_\alpha(P)(1,1)$ can be bounded by the linear function $\alpha c_1(p_1,p_2)$, where $c_1(p_1,p_2) = p_1|p_1+p_2-1|/\min\{p_2, p_2(p_1+p_2)\}$ is a (p_1,p_2)-dependent constant.

Furthermore, the asymptotic probabilities of going from one ergodic class to another (or to a transient state) are zero. This shows that $H_\alpha(P)$ uncovers the structure of the ergodic classes. In addition, the approximation assigns in general a positive mass to jumps from a transient state to itself, e.g.,

$$H_\alpha(P)(4,4) = \frac{\alpha\left(1 - \sum_{i=1}^{3} r_i\right)}{\alpha + (1-\alpha)\left(\sum_{i=1}^{3} r_i\right)},$$

which is strictly larger than zero if $\sum_{i=1}^{3} r_i < 1$, while clearly $\Pi_P(4,4) = 0$. Note that, besides the case $\sum_{i=1}^{3} r_i = 1$, the approximation may give the wrong impression that a transient state, say i, is ergodic. However, when α is chosen sufficiently small (for example of the order 10^{-8}), together with the structural property that there are no transitions from ergodic states towards i, the fact that i is transient becomes apparent.

In the following we analyse the effect that taking a power of the modified resolvent has on the convergence. Denote $(H_\alpha(P))^2(1,1)$ as the $(1,1)$-th element from $(H_\alpha(P))^2$. It then holds that

$$(H_\alpha(P))^2(1,1) = \frac{(\alpha(1-p_1) + (1-\alpha)p_2)^2 + p_1 p_2}{(\alpha + (1-\alpha)(p_1+p_2))^2},$$

the relative error of which can be computed to be equal to

$$\frac{|(H_\alpha(P))^2(1,1) - \Pi_P(1,1)|}{\Pi_P(1,1)} = \frac{\alpha^2 \left|(p_1+p_2)\left((1-p_1)^2 + p_1p_2\right) - p_2\right|}{p_2(\alpha + (1-\alpha)(p_1+p_2))^2},$$

which can be bounded by the quadratic function $\alpha^2 c_2(p_1,p_2)$ where

$$c_2(p_1,p_2) = \frac{\left|(p_1+p_2)\left((1-p_1)^2 + p_1p_2\right) - p_2\right|}{p_2(\min\{1, p_1+p_2\})^2}.$$

Note that when $p_1 + p_2 = 1$ or $p_1 = p_2 = 0$ the relative errors of approximation $H_\alpha(P)(1,1)$ and $(H_\alpha(P))^2(1,1)$ are zero. For $p_1 + p_2 \in (0,2] \setminus \{1\}$ and $\alpha \in (0,1)$ the relative error of $(H_\alpha(P))^2(1,1)$ is strictly smaller than that of $H_\alpha(P)(1,1)$. Furthermore, comparing the relative error bounds $\alpha c_1(p_1,p_2)$ and $\alpha^2 c_2(p_1,p_2)$ shows the quadratic improvement of $(H_\alpha(P))^2(1,1)$, which is in accordance with Theorem 5.1. This illustrates the improvement that can be achieved through the power k in the generalization. The other entries of $H_\alpha(P)$ can be analysed along the same lines. $\lhd$

In the following example we discuss the convergence of $H_\alpha(P)$ in case of a nearly decomposable Markov chain.

Example 5.3. In the light of nearly decomposable Markov chains it is interesting to see what happens in case $p_1 + p_2 \downarrow 0$ for the Markov chain in Example 5.1, i.e., when p_1 and p_2 are both close to 0. L'Hôpital's rule shows that the relative error of the $(1,1)$-th element for $p_1 + p_2 \downarrow 0$ in the limit equals

$$\frac{\alpha}{1-\alpha}\frac{p_1}{p_2},$$

see also the relative errors from Example 5.1. Similar, the relative error of $H_\alpha(P)(1,2)$ converges for $p_1 + p_2 \downarrow 0$ towards

$$\frac{\alpha}{1-\alpha}.$$

Both relative errors show that arbitrary accuracy can be achieved by using the modified resolvent with α small even in case of nearly decomposable Markov chains.

Now consider the case where $\sum_{i=1}^{3} r_i = \epsilon$, for $\epsilon > 0$ small. In that case the Markov chain almost breaks up into 3 ergodic classes. For the $(4,4)$-th element it holds that

$$H_\alpha(P)(4,4) = \frac{\alpha(1-\epsilon)}{\alpha + (1-\alpha)\epsilon},$$

which equals the absolute error, since $\Pi_{P_1}(4,4) = 0$. Choosing α such that

$$\alpha < \frac{\epsilon\delta}{1-\delta-(1-\delta)\epsilon}$$

leads to an absolute error smaller than δ, showing that arbitrary accurate precision can be achieved with $H_\alpha(P)$ even in case the Markov chain almost breaks up into 3 ergodic classes. Similar for the $(4,3)$-th element of $H_\alpha(P)$ it can be shown that the relative error equals

$$\frac{|H_\alpha(P)(4,3) - \Pi_P(4,3)|}{\Pi_P(4,3)} = 1 - \frac{\epsilon}{\alpha + (1-\alpha)\epsilon},$$

showing that in order to obtain a relative error smaller than η one should choose $\alpha \leq \epsilon\eta$, again showing that any accuracy can be achieved in theory. Note that $H_\alpha(P)(4,1) = H_\alpha(P)(4,2) = H_\alpha(P)(4,3) = 0$ in case $\sum_{i=1}^{3} r_i = 0$, i.e., in case the Markov chain consists of three ergodic classes this is correctly detected. $\lhd$

Alternatively to PM applied to $H_\alpha(P)$, one may compute the modified resolvent of $H_\alpha(P)$. More specifically, one may construct recursively a sequence $\{\overline{H}_\alpha(P;n) : n \in \mathbb{N}\}$ of *nested modified resolvents* with $\overline{H}_\alpha(P;0) = P$ and, for $n \geq 1$

$$\overline{H}_\alpha(P;n) = H_\alpha(\overline{H}_\alpha(P;n-1)).$$

As the following theorem shows, the norm error of $\overline{H}_\alpha(P^q;n)$ can be bounded by a geometric function with power n and rate α.

Theorem 5.2. *For $\alpha \in (0,1)$ such that $\alpha\gamma(P,q) < 1$ it holds that*

$$\|\Pi_P - \overline{H}_\alpha(P^q;n)\| \leq \frac{\gamma(P,q)\alpha^n}{1-\alpha\gamma(P,q)(1-\alpha^{n-1})}, \quad n \in \mathbb{N},$$

and

$$\lim_{n\to\infty} \overline{H}_\alpha(P^q;n) = \Pi_P.$$

Proof. Proof via mathematical induction. Because $\alpha\gamma(P,q) < 1$ it is clear that the bound holds true for $n = 1$ via Theorem 5.1. Now suppose it holds true for general $n - 1 \geq 1$, then

$$\|\Pi_P - \overline{H}_\alpha(P^q; n)\| = \|\Pi_P - H_\alpha(\overline{H}_\alpha(P^q; n-1))\|$$

since $\|(1-\alpha)\overline{H}_\alpha(P^q; n-1)\| < 1$ we can write out the inverse and bring Π_P inside summation

$$= \left\| \alpha \sum_{l=0}^{\infty} (1-\alpha)^l (\Pi_P - (\overline{H}_\alpha(P^q; n-1))^{l+1}) \right\|$$

straightforward bounding

$$\leq \alpha \sum_{l=0}^{\infty} (1-\alpha)^l \left\| \Pi_P - (\overline{H}_\alpha(P^q; n-1))^{l+1} \right\|$$

since $\Pi_P \overline{H}_\alpha(P^q; n-1) = \Pi_P$

$$\leq \alpha \sum_{l=0}^{\infty} (1-\alpha)^l \left\| \Pi_P - \overline{H}_\alpha(P^q; n-1) \right\|^{l+1}$$

filling in the induction hypothesis

$$= \frac{\gamma(P,q)\alpha^n}{1 - \alpha\gamma(P,q)(1-\alpha^{n-2})} \sum_{l=0}^{\infty} \left(\frac{(1-\alpha)\gamma(P,q)\alpha^{n-1}}{1 - \alpha\gamma(P,q)(1-\alpha^{n-2})} \right)^l$$

for all $n - 1 \geq 1$ when $\alpha\gamma(P,q) < 1$ it holds that $\frac{(1-\alpha)\gamma(P,q)\alpha^{n-1}}{1-\alpha\gamma(P,q)(1-\alpha^{n-2})} < 1$ and thus

$$= \frac{\gamma(P,q)\alpha^n}{1 - \alpha\gamma(P,q)(1-\alpha^{n-2}) - (1-\alpha)\gamma(P,q)\alpha^{n-1}}$$

taking out $\alpha\gamma(P,q)$ in the denominator gives

$$= \frac{\gamma(P,q)\alpha^n}{1 - \alpha\gamma(P,q)(1-\alpha^{n-1})}$$

thereby showing that it holds for n and thus ends the proof. □

Corollary 5.1. *For $\alpha \in (0,1)$ such that $\alpha\gamma(P,q) < 1$ it directly follows from the above theorem that*

$$\|\Pi_P - \overline{H}_\alpha(P; n)\| \leq \frac{\epsilon}{1-\epsilon}\alpha^{n-1}, \quad n \geq 1, \tag{5.14}$$

when we define $\epsilon = \alpha\gamma(P,q)$. Furthermore, since $\overline{H}_\alpha(P; n)\Pi_P = \Pi_P$ it holds for $k \geq 1$ that

$$\begin{aligned} \|\Pi_P - \left(\overline{H}_\alpha(P; n)\right)^k\| &= \|\left(\Pi_P - \overline{H}_\alpha(P; n)\right)^k\| \\ &\leq \left(\|\Pi_P - \overline{H}_\alpha(P; n)\|\right)^k \\ &= \left(\frac{\epsilon}{1-\epsilon}\alpha^{n-1}\right)^k, \end{aligned}$$

where in the last equation (5.14) *is used.*

Theorem 5.2 shows that repeated application of the modified resolvent yields a geometric improvement of the rate of geometric ergodicity. Example 5.4 illustrates Theorem 5.2.

Example 5.4. We revisit the instance from Example 5.1 and 5.2 where we assume that p_1, p_2, r_1, r_2 and r_3 are non-zero probabilities. For this chain $\overline{H}_\alpha(P;2)$ can be explicitly solved to be

$$\overline{H}_\alpha(P;2) = \begin{bmatrix} \frac{p_2+\alpha^2(1-p_1-p_2)}{p_1+p_2+\alpha^2(1-p_1-p_2)} & \frac{p_1}{p_1+p_2+\alpha^2(1-p_1-p_2)} & 0 & 0 \\ \frac{p_2}{p_1+p_2+\alpha^2(1-p_1-p_2)} & \frac{p_1+\alpha^2(1-p_1-p_2)}{p_1+p_2+\alpha^2(1-p_1-p_2)} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{(r_1+r_2)p_2(1-\alpha^2)+r_1\alpha^2}{((p_1+p_2)(1-\alpha^2)+\alpha^2)\xi(\alpha)} & \frac{(r_1+r_2)p_1(1-\alpha^2)+r_2\alpha^2}{((p_1+p_2)(1-\alpha^2)+\alpha^2)\xi(\alpha)} & \frac{r_3}{\xi(\alpha)} & \frac{(1-\sum_{i=1}^3 r_i)\alpha^2}{\xi(\alpha)} \end{bmatrix},$$

where $\xi(\alpha) := (1-\alpha^2)\sum_{i=1}^3 r_i + \alpha^2$. This shows that $\overline{H}_\alpha(P;2)$ converges with quadratic rate in terms of α towards Π_P. Consequently, a larger rate of convergence is achieved for $\overline{H}_\alpha(P;2)$ than for $H_\alpha(P)$; compare with Example 5.2. More specifically, the relative error of the $(1,1)$-th element equals

$$\frac{\left|\overline{H}_\alpha(P;2)(1,1) - \Pi_P(1,1)\right|}{\Pi_P(1,1)} = \left|\frac{p_1(1-p_1-p_2)}{p_2(p_1+p_2)+\alpha^2 p_2(1-p_1-p_2)}\right|\alpha^2,$$

where the absolute term converges to constant $\frac{p_1|1-p_1-p_2|}{p_2(p_1+p_2)}$ for α small. Similar, the relative error of $\overline{H}_\alpha(P;3)(1,1)$ equals

$$\frac{\left|\overline{H}_\alpha(P;3)(1,1) - \Pi_P(1,1)\right|}{\Pi_P(1,1)} = \left|\frac{p_1(1-p_1-p_2)}{p_2(p_1+p_2)+\alpha^3 p_2(1-p_1-p_2)}\right|\alpha^3,$$

showing that the relative error error is approximately a factor α smaller than that of $\overline{H}_\alpha(P;2)(1,1)$ in accordance with what can be expected from Theorem 5.2. ◁

Unfortunately, as $\gamma(P,q)$ is not available, it is neither clear what a good initial choice for α is, nor when to terminate $(H_\alpha(P))^k$ or the repeated application of $\overline{H}_\alpha(P;n)$. In the following we will address these two issues in more detail.

5.4 The Jump Start Power Method

In the previous section we have shown that going from P to the modified resolvent $H_\alpha(P)$ potentially yields a geometrically ergodic Markov chain with no transient phase (i.e., $r = 1$). In this section we show how this result can be made fruitful for numerical computations. In particular, Section 5.4.1 illustrates the modified resolvent theory through numerical experiments, Section 5.4.2 develops a practical method that exploits the developed theory by introducing the jump start power method (JSPM) and provides numerical results. Lastly, Section 5.4.3 discusses and numerically illustrates the use of JSPM in case of large (sparse) systems.

5.4.1 Motivating Numerical Experiments

As a first step we analyse the effect of mapping P to $H_\alpha(P)$ by comparing numerically P^n with $H_\alpha(P)$. The considered instances cover a wide range of Markov chains and for an overview we refer to Table 5.1. Each row in Table 5.1 corresponds to an instance

defined by its transition matrix (Tr. Matrix). Most instances are based on random graph models that capture key properties of real-life networks. The instances vary in terms of size S (given in the 'S' column), structure (given in the 'Ergodic Structure' column), connectivity (as indication, column '$p^\star$' gives the smallest non-zero element of P), and parameters used for random graph models (given in the 'Description' column). The ergodic structure is denoted by $([v_1, v_2, \cdots, v_E], \mathcal{T})$, where E is the number of ergodic classes, v_i the number of states in the i-th ergodic class and $\mathcal{T} = S - \sum_{i=1}^{E} v_i$ is the number of transient states. In case of transient states, the corresponding part in P is randomly filled such that transient states most likely point towards each other and multiple ergodic classes (if possible). The description column gives the relevant reference of the instance together with parameters given in the same order as as they appeared in the original reference, where altered labels are used in case of conflicting notation (e.g., if α is used as parameter in the original reference, we refer to this parameter as β). For the implementation of the different instances the code provided in MATLAB toolbox CONTEST [178] was used. The CONTEST toolbox generates symmetric adjacency matrices which are in many cases periodic. In order to obtain the corresponding transition matrix P, the rows are first normalized ensuring that each row sums up to one. Afterwards, the transition matrix P is mixed with the identity matrix to achieve aperiodicity, which does not affect the ergodic behaviour of the chain.

Comparison of P^n and $H_\alpha(P)$: In the first numerical experiment, we compute for a series of Markov chains P_i, with $1 \leq i \leq 4$ in Table 5.1, the power $n_\alpha(P)$ such that

$$\left| \|\Pi_P - H_\alpha(P)\| - \|\Pi_P - P^{n_\alpha(P)}\| \right| < 10^{-12}.$$

In words, $n_\alpha(P)$ is the power of P that is substituted by $H_\alpha(P)$. Note that a power $n_\alpha(P)$ can be obtained via $\log_2 n_\alpha(P)$ matrix multiplications. The numerical results depicted in Figure 5.1 show that the modified resolvent can replace PM approximations for large powers.

In particular for the Courtois matrix P_1, in order to approximately achieve a norm error of $7.92 \cdot 10^{-7}$ a power is needed of 2^{16} while the same norm error is obtained via the modified resolvent with $\alpha \approx 10^{-10}$. For P_4 the modified resolvent with $\alpha \approx 10^{-11.18}$ leads to the same norm error (of approximately $1.63 \cdot 10^{-5}$) as PM with power 20655175 ($\approx 2^{24.3}$). As for computation times, experiments showed that on average PM $(P_4)^k$ with power $k = 20655175$ takes on average 73.12 seconds in a sparse matrix setting whereas the modified resolvent $H_{\alpha=10^{-11.18}}(P_4)$ takes on average 2.68 seconds, i.e., a difference of factor 27.28 on average[5].

Length of transient phase for $H_\alpha(P)$: Figure 5.2 illustrates the effect that powers of $H_\alpha(P)$ have on the norm error for the Courtois matrix. Different values for α are considered and for each α the exponential decay location is determined and thereby the length of the transient phase is identified. Note that $H_{\alpha=1}(P)$ equals P. A heuristic approach is used to find the exponential decay location where for each α an exponential function is repeatedly fitted to the data until the coefficient of determination R^2 is close enough to 1 (where $R^2 = 1$ represents a perfect fit). After each fit which led to an insufficient coefficient of determination, the dataset is reduced by increasing the value

[5] The experiments were performed in MATLAB R2011b on a 64-bit Windows desktop PC with Intel(R) Core(TM) i5-2310 CPU @ 2.90GHz processor

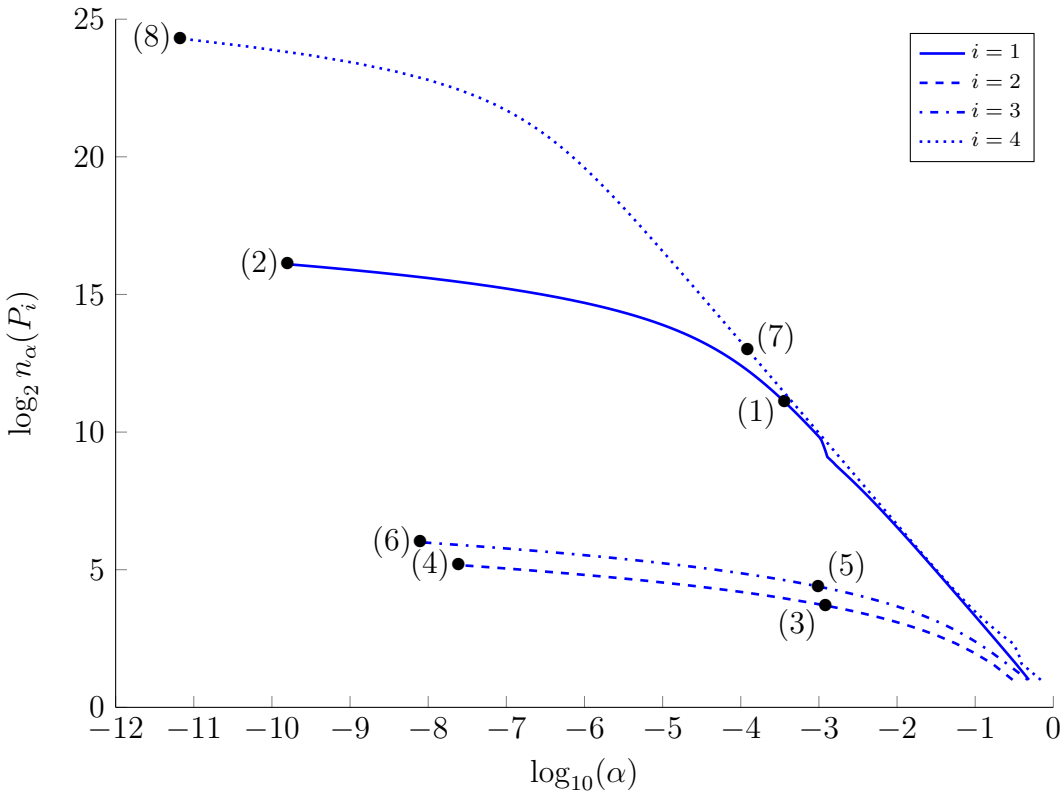

Fig. 5.1: Combinations of α and $n_\alpha(P_i)$ such that $\left|\|\Pi_{P_i} - H_\alpha(P_i)\| - \|\Pi_{P_i} - (P_i)^{n_\alpha(P_i)}\|\right| < 10^{-12}$ for $i = 1, 2, 3, 4$. Some specific points (•) are highlighted which have the following ∞-norm errors, resp.: (1) $\to$ 0.339, (2) $\to$ 7.92e−7, (3) $\to$ 0.000345, (4) $\to$ 7.3e−8,(5) $\to$ 0.00126, (6) $\to$ 5.2e−8, (7) $\to$ 1.58, (8) $\to$ 1.63e−5.

of the first considered power k and the fitting repeats. The found exponential decay locations (i.e., the smallest power in the dataset that led to R^2 sufficiently close to 1) are denoted with the large dots and labelled, where the labels correspond to the fitted functions given in the caption together with the R^2 in parenthesis behind the function.

The main observation from Figure 5.2 is that the exponential decay locations shift to the left (and thereby the transient phase becomes smaller) for decreasing values of α.

This phenomenon has been theoretically shown in the previous section. It is worth noting for $\alpha \leq 10^{-3}$ there is no transient phase, i.e., $r = 1$ in these cases. Furthermore, from the fitted functions it follows that smaller α values lead to stronger norm error reduction for increasing powers.

The nested modified resolvent $\overline{H}_\alpha(P; n)$: Similar to Figure 5.2, Figure 5.3 illustrates the effect of the nested modified resolvent $\overline{H}_\alpha(P; n)$ for varying n in case $P = P_1$.

It shows that relative large values for α already lead to small norm errors after few iterations. In particular, the fitted relation between the norm error of $\overline{H}_{\alpha=0.01}(P; n)$ and n is approximately $4901e^{-4.6n}$ whereas that of $(H_{\alpha=0.01}(P))^n$ and n is approximately $e^{-0.0198n}$ (see also Figure 5.2), showing that the effect of an increase in the number of iterations in the nested modified resolvent is far more effective than an increase in the power of the modified resolvent for the same α. It therefore illustrates the

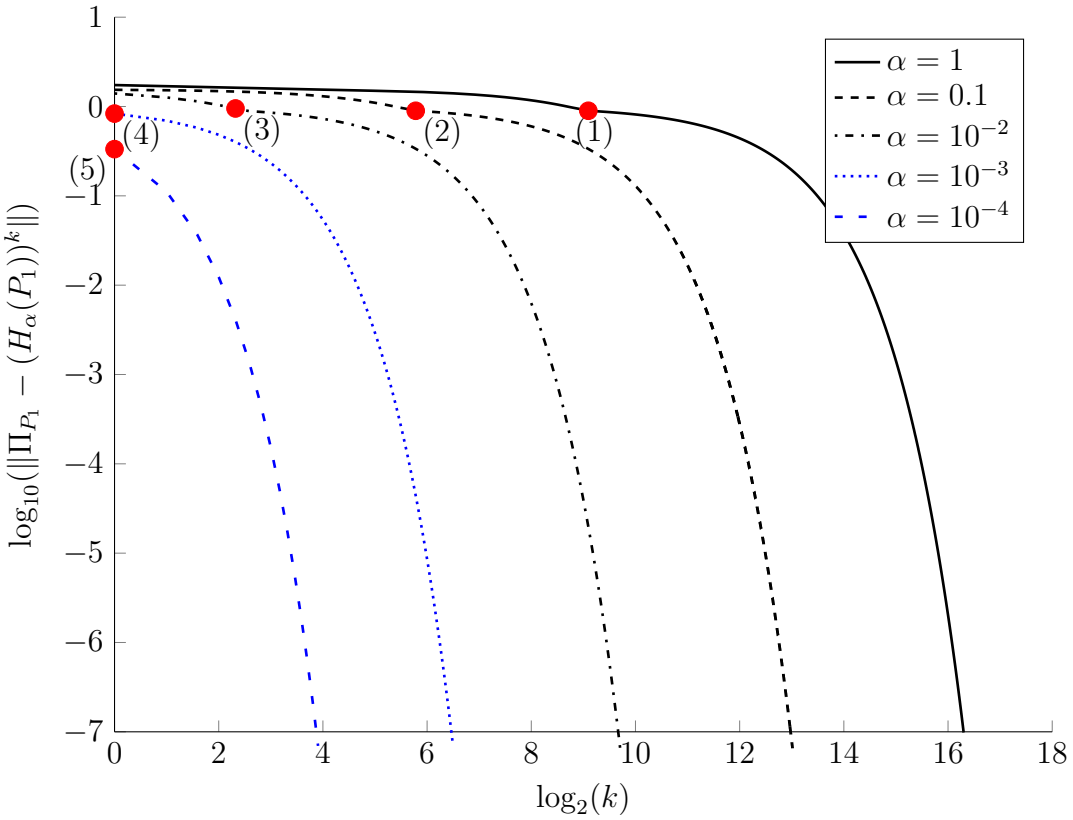

Fig. 5.2: Development norm errors for powers of modified resolvent in case of the Courtois matrix. The fitted functions from exponential decay locations indicated by • are (with coefficient of determination R^2 in brackets behind):
(1) $\to 1{\cdot}e^{-0.00020002\cdot k}$ (1), (2) $\to 1{\cdot}e^{-0.0019984\cdot k}$ (1), (3) $\to 1{\cdot}e^{-0.019807\cdot k}$ (1),
(4) $\to 1{\cdot}e^{-0.18235\cdot k}$ (1), (5) $\to 1{\cdot}e^{-1.0987\cdot k}$.

sharper bound found for the nested modified resolvent in comparison with powers of the modified resolvent.

5.4.2 Development of the Jump Start Power Method (JSPM)

In this section we will develop a power-method like algorithm based on the theory established in the previous section. To that end we discuss how to choose α and we provide a stopping rule for the algorithm. Our recommendations are based on numerical experiments and balance avoiding numerical issues with achieving good numerical approximations.

Numerical experiments indicate that in order to achieve high accuracy it is best to choose relative large α and take powers of the resulting modified resolvent. Unfortunately, the modified resolvent is not a sparse matrix and computing powers is rather costly. Alternatively, one may choose α (very) small and calculate the modified resolvent only once (without taking powers). This, however, may lead to numerical issues. For illustration purposes consider the Courtois matrix, i.e., P_1, for which it holds that

$$\|\Pi_{P_1} - H_{\alpha=10^{-10}}(P_1)\| \approx 5.09 \cdot 10^{-7}.$$

According to the theoretical results decreasing α to, say, $\alpha = 10^{-12}$ should improve the

Tr. Matrix	**Description**	S	**Ergodic Structure**	$p^\star$
P_1	Courtois matrix, [173].	8	$([8], 0)$	3e-5
P_2	Kleinberg's network, [123], with parameters $p = 5$, $q = 2$ and $\beta = 3$.	225	$([225], 0)$	1e-2
P_3	Kleinberg's network, [123], with parameters $p = 3$, $q = 6$ and $\beta = 1$.	900	$([900], 0)$	1e-2
P_4	Block diagonal transition matrix of P_2 and P_3 with weak connection between P_2 and P_3. I.e., two random nodes from P_2 and P_3, resp., are connected with probability 1e-4.	1125	$([1125], 0)$	1e-4
P_5	Block diagonal transition matrix of P_2 and P_3 with 60 transient states.	1185	$([225, 900], 60)$	8.4e-6
P_6	Block diagonal transition matrix of P_1 and P_2 with 20 transient states.	253	$([8, 225], 20)$	3e-5
P_7	RENGA protein-protein interaction (PPI) network, [80], with parameters $\beta_1 = 0.4$ and $\beta_2 = 0.965$.	700	([202, 181, 67, 98, 152],0)	1.7e-3
P_8	Preferential attachment network, [24], with parameter $d = 5$.	700	([700],0)	4.7e-5
P_9	Lock and key PPI network, [152], with parameters $\beta_1 = 0.4$ and $\beta_2 = 0.965$.	700	([700],0)	1.7e-4

Tab. 5.1: Instances used for numerical experiments. For implementation the MATLAB code provided by CONTEST [178] was used.

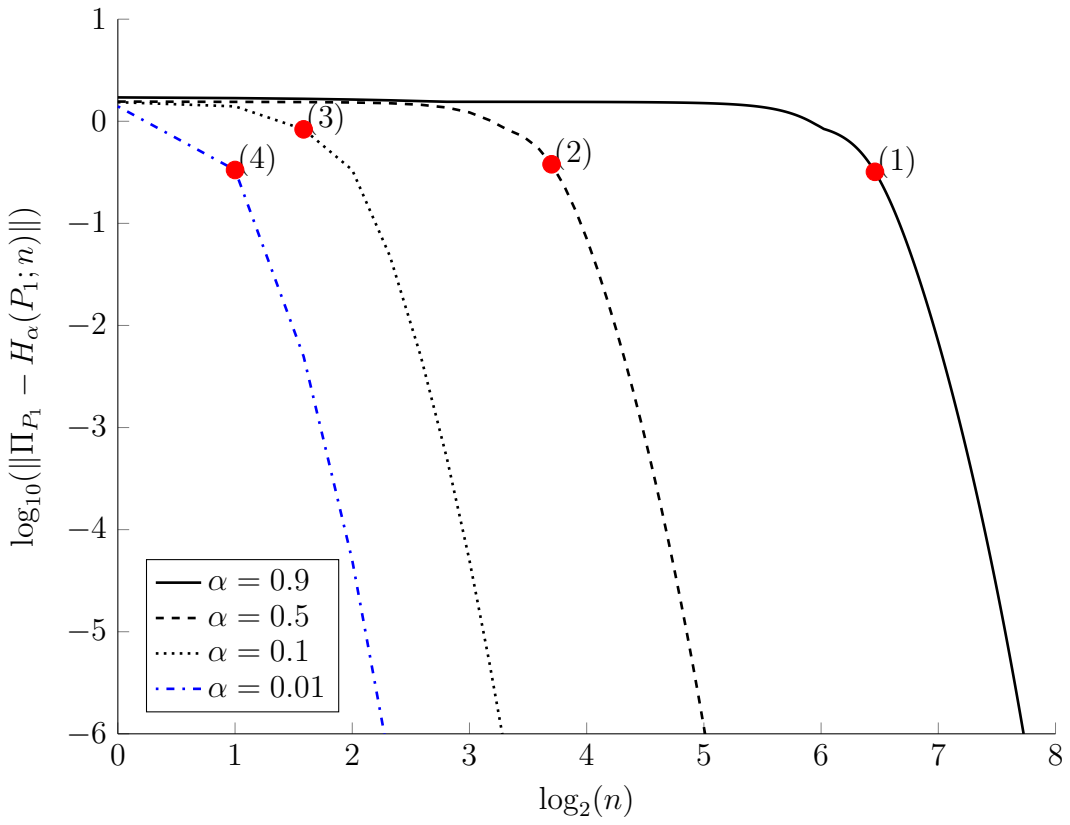

Fig. 5.3: Development norm errors for nested modified resolvent in case of the Courtois matrix. The fitted functions from exponential decay locations indicated by • are (with coefficient of determination R^2 in brackets behind):
(1) $\to 926.3877 \cdot e^{-0.090026 \cdot n}$ (0.99903), (2) $\to 1122.4366 \cdot e^{-0.60497 \cdot n}$ (0.9995),
(3) $\to 835.0589 \cdot e^{-1.9565 \cdot n}$ (0.99996), (4) $\to 4901.0117 \cdot e^{-4.6002 \cdot n}$ (1)

quality of the approximation (see Theorem 5.1). But the contrary is true as

$$\|\Pi_{P_1} - H_{\alpha=10^{-12}}(P_1)\| \approx 3.48 \cdot 10^{-5}.$$

This shows that numerical issues come into play when computing the modified resolvent for $\alpha = 10^{-12}$ for P_1. A similar effect can be observed for the nested modified resolvent

$$\|\Pi_{P_1} - \overline{H}_{\alpha=10^{-3}}(P_1;3)\| \approx 5.00 \cdot 10^{-6} \quad \text{compared to} \quad \|\Pi_{P_1} - \overline{H}_{\alpha=10^{-4}}(P_1;3)\| \approx 3.62 \cdot 10^{-5},$$

and even for PM with P_1, i.e.,

$$\|\Pi_{P_1} - P_1^{10^5}\| \approx 2.06 \cdot 10^{-9} \quad \text{compared to} \quad \|\Pi_{P_1} - P_1^{10^{10}}\| \approx 1.22 \cdot 10^{-5}.$$

In Figure 5.4 the norm errors of $(H_\alpha(P_1))^k$, i.e., $\|\Pi_{P_1} - (H_\alpha(P_1))^k\|$, are plotted for varying $\alpha \in (0,1)$ and powers k. From Figure 5.4 it follows for each k that choosing α too small leads to numerical issues and consequently leading to an increase of norm errors, contrary to what can be expected from the theory. E.g., for $k = 1$ numerical issues appear when choosing α smaller than 10^{-10} from where the norm errors start increasing in a zigzag pattern. Furthermore, the figure shows that the smallest norm errors can be achieved by larger powers k and relative larger α.

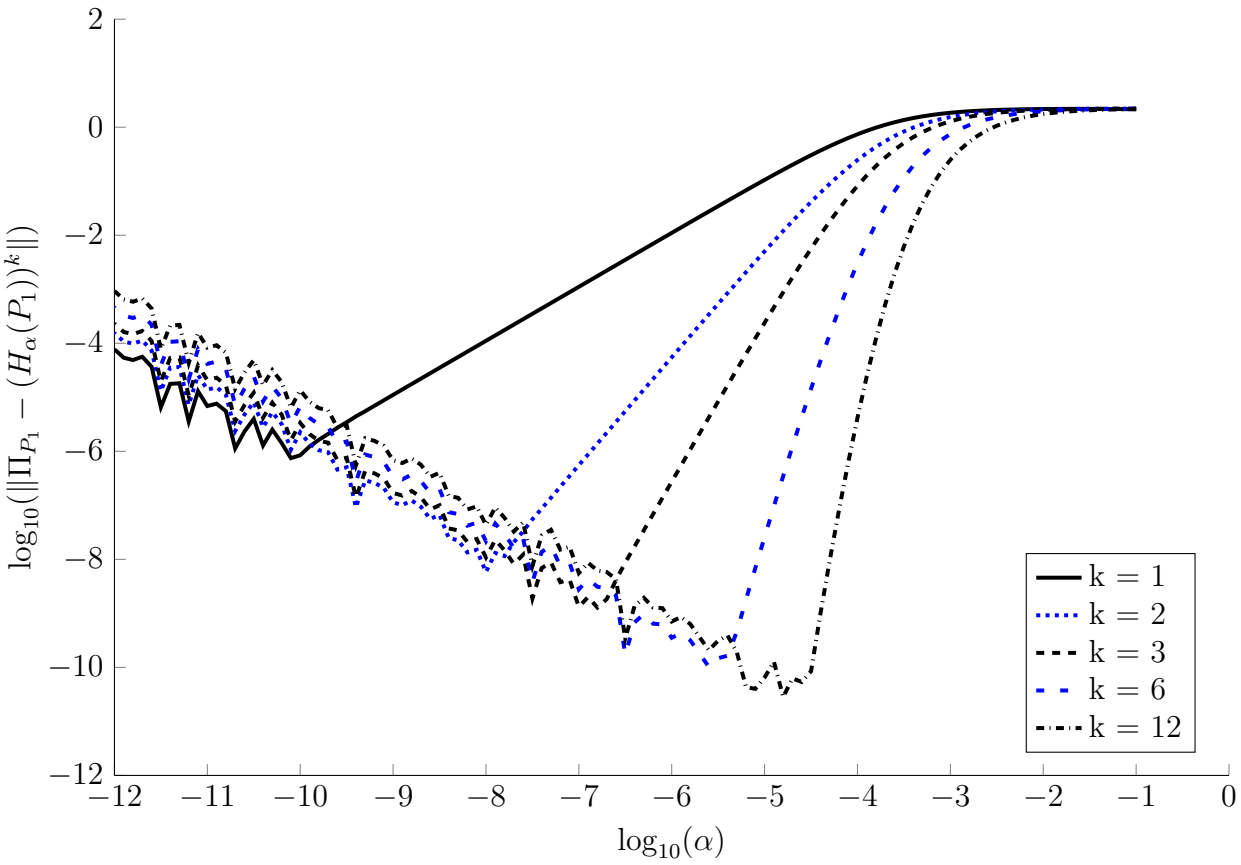

Fig. 5.4: Illustration of the norm errors for $(H_\alpha(P_1))^k$ with varying α and k.

Based on the above results, we advice to use the modified resolvent in a PM framework with a carefully chosen α. When to terminate the power iterations is a delicate matter. A natural stopping rule is to terminate the algorithm when the improvement of an extra iteration becomes insignificant. More specifically, in order to find a power k such that $\|\Pi_P - P^k\| \leq \epsilon$, one may terminate PM if $\|P^k P - P^k\| < \epsilon$, for $\epsilon > 0$ small. Unfortunately, this stopping rule may stop the algorithm too early as is illustrated in Example 5.5 below.

Example 5.5. For $\delta \in (0,1)$ and $k \in \mathbb{N}$ let

$$P = \begin{bmatrix} 1-\delta & \delta \\ 0 & 1 \end{bmatrix} \quad \text{so that} \quad \Pi_P = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad P^k = \begin{bmatrix} (1-\delta)^k & 1-(1-\delta)^k \\ 0 & 1 \end{bmatrix}.$$

It can be shown that $\|P^k P - P^k\| = 2\delta(1-\delta)^k$ and $\|\Pi_P - P^{k+1}\| = 2(1-\delta)^k$, which implies that

$$\delta \|\Pi_P - P^{k+1}\| = \|P^k P - P^k\|.$$

Hence, $\|P^k P - P^k\| \leq \epsilon$ only implies

$$\|\Pi_P - P^{k+1}\| \leq \frac{\epsilon}{\delta},$$

which for small values of δ (e.g. $\delta < \epsilon/2$) provides no insight and thereby showing that the stopping rule is insufficient. ◁

To prevent a similar pitfall when using $\|(H_\alpha(P))^k H_\alpha(P) - (H_\alpha(P))^k\| < \epsilon$ for $\alpha \in (0,1)$ as stopping rule, we recommend choosing α such that

$$\alpha = \min\{\alpha_{\max},\ p^\star/N\}, \tag{5.15}$$

where $\alpha_{\max}$ is a user-defined upper bound for α, $p^\star$ denotes the minimal non-zero value of P, formally given by,

$$p^\star := \min_{i,j}\{P(i,j) : P(i,j) > 0\},$$

and N is a user-defined scaling to ensure α is significantly smaller than $p^\star$. The intuition is that by choosing $\alpha \ll p^\star$ the effect of the smallest transition is taken into account. As illustrated by our numerical examples, even for a nearly decomposable matrix, the minimal non-zero entry is typically not so small that α given in (5.15) leads to numerical instabilities for $H_\alpha(P)$. In addition, choosing α as in (5.15) typically leads to $H_\alpha(P)$ having no transient phase (i.e., $r = 1$).

The above considerations lead to the following *jump start power method* (JSPM).

(1) Choose $\alpha = \min\{\alpha_{\max},\ p^\star/N\}$ and select numerical precision ϵ.

(2) Initialize $k = 1$ and calculate $H_\alpha(P)$.

(3) Set $k = k + 1$.

(4) If
$$\|(H_\alpha(P))^{k-1} H_\alpha(P) - (H_\alpha(P))^{k-1}\| \geq \epsilon$$
go to step 3. Otherwise go to step 5.

(5) Return $(H_\alpha(P))^k$.

It is worth noting that rather than computing the resolvent in Step 2 of the JSPM directly, which requires evaluating the inverse of $(1-\alpha)P$, it is numerically more efficient to solve $(I - (1-\alpha)P) = \alpha P$, which reduces the problem to that of solving systems of linear equations.

In Table 5.3 some numerical results for JSPM are shown. For an overview of the instances see Table 5.1. Two parameter choices for α and ϵ are considered, see Table 5.2. Parameter Setting 1 aims at achieving higher accuracy of the approximation (i.e., small value for ϵ), which is numerical possible by choosing α not too small. Parameter Setting 2 focuses more on quick convergence of the algorithm, i.e., a larger value for ϵ compared to setting 1 and small α.

From the results put forward in Table 5.3 it follows that significantly smaller norm errors can be achieved using Parameter Setting 1 but at the cost of more iterations (read, powers of $H_\alpha(P)$). Since the modified resolvent is typically not sparse taking powers of $H_\alpha(P)$ becomes impractical for large Markov chains. This issue will be topic of next subsection.

	α	ϵ	**Aim**
Parameter Setting 1:	$\min\{10^{-4},\ p^\star/100\}$	10^{-8}	high accuracy
Parameter Setting 2:	$\min\{10^{-8},\ p^\star/S\}$	10^{-6}	fast computation

Tab. 5.2: Parameter settings used for JSPM in numerical experiments.

	Parameter Setting 1		*Parameter Setting 2*	
Tr. Matrix	**Norm Error**	**# Iterations**	**Norm Error**	**# Iterations**
P_1	2.4461e-010	4	4.8780e-009	3
P_2	4.3119e-012	4	1.8567e-008	2
P_3	1.1869e-012	4	1.0155e-008	2
P_4	1.6029e-008	42	1.3002e-008	5
P_5	4.3520e-009	3	1.9519e-008	2
P_6	5.3403e-010	4	2.5750e-008	3
P_7	4.7504e-010	8	5.9758e-009	3
P_8	2.4177e-010	3	1.6035e-008	2
P_9	4.6657e-010	3	5.9593e-008	2

Tab. 5.3: Results of JSPM for two parameter settings given in Table 5.2.

5.4.3 JSPM for Large Markov Chains

This final subsection discusses JSPM for large Markov chains. A common feature of large chains is that the transition matrix P is sparse but the ergodic projector Π_P is not due to connectivity [157]. This leads to numerical issues in approximating Π_P. In particular for JSPM: when the approximation $(H_\alpha(P))^k$ approaches Π_P as k is increasing, iterations become computational more expensive and a memory burden emerges due to the loss of sparsity.

Therefore, in case of large instances, our (on numerical experiments based) advice is to choose α significantly small and return $H_\alpha(P)$ as approximation, i.e., apply the JSPM for $k = 1$. In addition, instead of calculating $H_\alpha(P)$ as a whole, we recommend to calculate a concentrated version of $H_\alpha(P)$, denoted by $H^c_\alpha(P)$, where the computation of $H^c_\alpha(P)$ elaborates structural properties of Π_P such as the fact that all rows corresponding to ergodic states from the same ergodic class are identical. In particular, when row i of $H_\alpha(P)$, denoted by $H_\alpha(P)(i,\cdot)$, is calculated, then based on this approximation it can be decided whether i is ergodic or transient by inspecting the value of $H_\alpha(P)(i,i)$. Indeed evoking the diagonal criterion, see Section 5.2, state i is ergodic if and only if $H_\alpha(P)(i,i)$ is significantly larger than 0. In case i is identified as ergodic, all indices corresponding to (significantly) positive entries of $H_\alpha(P)(i,\cdot)$ are identified as belonging to the same ergodic class. Vector $H_\alpha(P)(i,\cdot)$ is saved in $H^c_\alpha(P)$ as approximation for the rows of the particular ergodic class and we are done considering all the indices from this ergodic class. In case i is identified as transient, $H_\alpha(P)(i,\cdot)$ is saved in $H^c_\alpha(P)$ as approximation for the i-th row of Π_P. We will refer to this procedure as the *adapted JSPM* version for large instances.

In the following we introduce the adapted JSPM algorithm, where $\mathcal{E}_j$ denotes the set of indices identified as part of the j-th ergodic class, E denotes the number of identified

Tr. Matrix	Description	S	Ergodic Structure	$p^\star$
P_{10}	Preferential attachment network, [24], with parameter $d = 2$.	10000	([10000],0)	1.1e-3
P_{11}	Kleinberg's network, [123], with parameters $p = 1$, $q = 1$ and $\beta = 1.5$.	15625	([15625], 0)	4.6e-2
P_{12}	Block diagonal transition matrix of P_{10} and P_{11} with weak connection between P_{10} and P_{11}. I.e., two random nodes from P_{10} and P_{11}, resp., are connected with probability 1.5e-9.	25625	([25625],0)	1.5e-9
P_{13}	Block diagonal transition matrix of P_{10} and P_{11} with 30 transient states.	25655	([10000 15625],30)	1.9e-6

Tab. 5.4: Large instances used for numerical experiments. It has the same setup as Table 5.1, see also the corresponding description in Section 5.4.1. For implementation the MATLAB code provided by CONTEST [178] was used.

ergodic classes, and C denotes the set of considered/evaluated states/indices. The user-defined value to decide whether a state is ergodic or not is denoted by ι, with $\iota > 0$. For large instances P the adapted JSPM for computing $H^c_\alpha(P)$ can be summarized as follows (recall that $\mathbb{S}$ is the state space of the Markov chain under consideration):

(1) Choose $\iota > 0$.

(2) Initialize $E = 0$ and $C = \emptyset$.

(3) If $\mathbb{S} \setminus C \neq \emptyset$:

  (3.1) Select $i \in \mathbb{S} \setminus C$.

  (3.2) Calculate $H_\alpha(P)(i, \cdot)$.

  Otherwise go to step 6.

(4) If $H_\alpha(P)(i, i) > \iota$:

  (4.1) State i is identified as ergodic, set $E = E + 1$.

  (4.2) $\mathcal{E}_E = \{j \ : \ H_\alpha(P)(i, j) > \iota\}$.

  (4.3) $C = C \cup \mathcal{E}_E$.

  Otherwise i is identified as transient, set $C = C \cup \{i\}$.

(5) Save $H_\alpha(P)(i, \cdot)$ in $H^c_\alpha(P)$ and go to step 3.

(6) Return $H^c_\alpha(P)$.

	$\|\Pi_{P_i} - \boldsymbol{H}^c_\alpha(\boldsymbol{P_i})\|$	**Computation Time (in sec.)** $\boldsymbol{H}^c_\alpha(\boldsymbol{P_i})$
$i = 10$	2.0553e-010	0.24
$i = 11$	1.6724e-010	1.14
$i = 12$	3.5243e-008	1.37
$i = 13$	6.1897e-006	45.80

Tab. 5.5: Results for adapted JSPM, with $\alpha = \min\{10^{-10},\ (p^\star)^2\}$ and $\iota = (1/S)^2$, in case of large instances.

For instances P_{10}, P_{11}, P_{12} and P_{13} from Table 5.4 the adapted JSPM is applied where we have chosen $\alpha = \min\{10^{-10},\ (p^\star)^2\}$ and $\iota = (1/S)^2$. The philosophy behind the choice of α is similar to Parameter Setting 2 in the previous section, i.e., small α is chosen such that one iteration is most likely sufficient. Our experience for real life networks is that when choosing $\iota = (1/S)^2$ correct distinctions are made between transient and ergodic states. In Table 5.5 the norm errors and computation times in seconds (sec.) of the experiments can be found.

From the results it follows that high accuracy is achieved in a relative small amount of time, i.e., an unique row of $H_\alpha(P)$ in case of 25625 states is calculated with MATLAB R2011b in 1.4 seconds on a 64-bit Windows desktop PC with Intel(R) Core(TM) i5-2310 CPU @ 2.90GHz processor with norm error $3.524 \cdot 10^{-8}$. To put the results into context, for instance P_{12} it takes on average 2.49 seconds to calculate $\mu^\top (P_{12})^4$ with norm error $\|\mu^\top \Pi_{P_{12}} - \mu^\top (P_{12})^4\| = 0.1794$, where $\mu^\top$ equals the first row of an appropriate sized identity matrix. It becomes even more counterproductive if we calculate $\mu^\top (P_{12})^8$, which takes on average 593.75 seconds and leads to norm error $\|\mu^\top \Pi_{P_{12}} - \mu^\top (P_{12})^8\| = 0.079$. The significant increase in computation time is due to loss of sparsity. Similar observations can be expected for the other large instances.

A way to (most likely) improve accuracy of the adapted JSPM without significantly increasing computation time is to calculate $H^c_\alpha(P^q)$, for $q > 1$. The intuition is that for relatively small q, P^q may not affect the sparsity too much (increase in computation time is limited) but may increase the accuracy (which is likely according to the theory). Note that although it is common, it is not necessary that larger q increases accuracy, theory only provides upper bounds for the norm error. Example 5.6 provides an instance for which a larger q does not increase accuracy of $H_\alpha(P)$.

Example 5.6. Take

$$P = \begin{bmatrix} 0.1 & 0.45 & 0.45 \\ 0.9 & 0.1 & 0 \\ 0.9 & 0 & 0.1 \end{bmatrix}$$

which gives

$$\Pi_P = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/4 \\ 1/2 & 1/4 & 1/4 \end{bmatrix} \quad \text{and} \quad P^2 = \begin{bmatrix} 0.82 & 0.09 & 0.09 \\ 0.18 & 0.415 & 0.405 \\ 0.18 & 0.405 & 0.415 \end{bmatrix}.$$

It holds that $\|\Pi_P - H_{\alpha=10^{-6}}(P)\| \approx 4.44 \cdot 10^{-7}$, whereas $\|\Pi_P - H_{\alpha=10^{-6}}(P^2)\| \approx 1.78 \cdot 10^{-6}$, which is due to the periodic behaviour of P (visible by comparing P and P^2). $\lhd$

For the instances put forward in Table 5.4 we tested the effect of considering $(P_i)^2$ instead of P_i, where $i = 10, 11, 12, 13,$. Most of the time taking a power increased the accuracy but more significantly increased computation time so that practical usability of taking powers is questionable.

5.5 Conclusion

This chapter introduces JSPM which is a generalization of power method (PM). JSPM is a highly accurate approximation method for the ergodic projector of a general aperiodic finite Markov chain. Convergence analysis and numerical experiments show that it can provide a viable generalization of PM. Especially in the case of large-scale Markov chain modelled systems JSPM works well and can deal with nearly decomposable chains without running into numerical instabilities.

Further research will try to achieve higher accuracy via numerical ingenuity. Also it will try to answer the question whether the approximation method also directly applies to periodic Markov multi-chains as numerical experiments suggests. The next chapter elaborates JSPM techniques to the approximation of the in Markov chain theory fundamental deviation matrix.