Stockholm University

Department of Physics

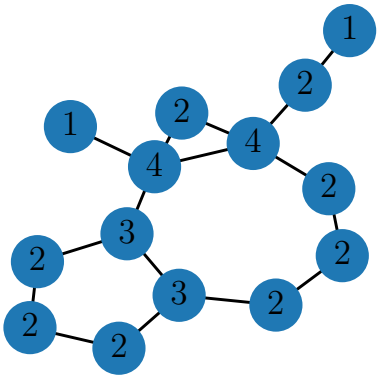
NORDITA, Nordic Institute for Theoretical Physics

Adaptive random walks on graphs to sample rare events

Master Thesis in Computational Physics

May 2023

Author:	David Christoph Stuhrmann
1st Supervisor:	Francesco Coghi
2nd Supervisor:	Supriya Krishnamurthy
Examiner:	Markus Kowalewski



Abstract

In this thesis, I study fluctuations and rare events of time-additive observables of discrete-time Markov chains on finite state spaces. The observable of interest is the mean node connectivity visited by a random walk running on instances of an Erdős-Rényi (ER) random graph. I implement and analyze the Adaptive Power Method (APM) which converges to the driven process, a biased random walk defined through a control parameter that simulates trajectories corresponding to rare events of the observable in the original dynamics. The APM demonstrates good convergence and accurately produces the desired quantities from a single trajectory. Due to the bulk-dangling-chain structure in the ER graph, the driven process seems to undergo a dynamical phase transition (DPT) for infinitely large graphs, meaning the behavior of the trajectories changes abruptly as the control parameter is varied. Observations show that the random walk visits two distinct phases, being de-localized in the bulk or localized in the chain. Through two simpler models capturing the bulk-dangling-chain property of the ER graph I study how the DPT occurs as the graph size increases. I observe that the trajectories of the driven process near the transition show intermittent behavior between the two phases. The diverging time scale of the DPT is found to be the average time that the random walk spends in a phase before it transitions to the other one. On the ER graph the trajectories are also intermittent but the form of the time scaling remains open due to computational limits on the graph size.

Contents

Introduction

In Isaac Asimov's Foundation series Golan Trevize traveled through the galaxy searching from planet to planet in a seemingly random fashion. But he ended up finding what he looked for: Earth. Often we are constrained on the paths that we take, for example that we only drive our cars on roads or that Golan Trevize gets advise on which planet to visit next only one at a time. With no prior knowledge one can only choose randomly from the paths and after very long times will eventually visit every possible place. Fortunately, there is a sense of a *better* randomness if one wants to find a specific place or path. This is like taking educated guesses and observing that after some time many small steps in the right direction add up to where one wishes to go. In this manner random walks on graphs can be tweaked to find rare events, like finding Earth in the vastness of the galaxy.

In this thesis, I focus on an adaptive random walk running on a random graph. The random walk exhibits its adaptive nature by actively steering itself towards sampling rare events. This is critical for understanding how the rare events are realized in time.

Loosely speaking, graphs are collections of points connected by lines. They are used as a mathematical tool to describe network structures found in both man-made and natural contexts. Human-made networks include the internet, transportation networks and information networks, e. g. citation networks, while social networks describe various interactions between humans or agents. In nature, examples are chemical reaction networks, neural networks and food webs [1–3]. Although graphs do not require to be embedded into a physical space, there exist metrics defining the shortest paths, walks and other spacial quantities [3]. One can therefore use the intuition built on spacial setups also for abstract graphs. The ability of graphs to capture a diversity of structures makes them a versatile tool for the study of networks.

Graphs are also utilized as the underlying state space of dynamical processes. The dynamics for some of these processes is given by a random walk, where the walker jumps from one node to another according to a set of rules. Random walks based on Markov chains, which do not have memory, are easy to implement and can even be studied analytically. This makes them a widely applicable tool to study physical systems where one only focuses on the interactions between the nodes. Examples of such processes are infection spreading, traffic, searching on networks and diffusion [1–3]. The network structure may also be changed by a dynamical process embedded in it, for example percolation where links are created and destroyed over time [1]. In statistical physics one is often interested in a set of internal micro-states of a system, e. g. a gas or fluid made up of many particles, that is represented by a graph and where the random walker transitions between these states. The long time behavior of the random walk then gives insights into the emergence and stability of macro-states [4, 5].

Along the random walk one can define time-additive observables which may represent statistical mechanics quantities such as energy, work or entropy production associated with the

steps taken [6]. For large times the observables converge to their typical value. But as we can only study random walks for a finite time this intrinsically leads to fluctuations in the value observed. Large Deviation Theory is the right framework to look at the probability distribution of these fluctuations [5, 7]. It describes the probability distribution in a logarithmic scale and thus allows to estimate the probabilities of very unlikely observable outcomes, called rare events. Although rare events occur with very low probabilities their strong effects dominate and lead the future dynamics of the system whenever they happen. A severe example are mass-extinction events in the Earth's history where a majority of species dies out over a very short period of time. Given the large deviation quantities one can construct a driven or auxiliary process, which is a modified random walk. The typical value of the driven process is a rare event of the original Markov process which would otherwise be exponentially hard to sample directly [7, 8]. The driven process is steered into sampling the desired rare event by a single external control parameter thus enabling one to study how these events occur.

A numerical method in finding the driven process is based on a stochastic control scheme which was first implemented by Borkar et al. [9] in the context of modelling a network of communication links. Based on that, the Adaptive Power Method algorithm formulated by Coghi and Touchette [10] is an adaptively learning random walk which converges to the driven process. The algorithm was implemented for continuous time and space Markov chains and, especially, it allows to study non-equilibrium systems modelled by Markov chains [11].

In my thesis I implement the Adaptive Power Method algorithm by Coghi and Touchette [10] in discrete space and time. The method allows to compute the large deviation functions of the mean degree observable and sample rare events. I consider Erdős-Rényi random graphs which are easy to construct, widely known, and often used as a playground to understand structure and dynamics of more complicated networks [6, 12, 13]. The benefit of the adaptive power method is that it can even efficiently sample rare events on networks where the global structure is unknown. Further, the analysis of the algorithm indicates that the driven process on the Erdős-Rényi graph might show a first order dynamical phase transition for infinitely large graphs. This means that dynamically in the trajectories leading to a particular fluctuation of the mean degree two different phases corresponding to the bulk and the chain of the graph are observed. By explicitly constructing the driven process and looking at its trajectories I study what happens for bigger and bigger graphs, potentially approaching the dynamical phase transition. The association is that a dynamical phase transition coincides with a diverging time scale [14]. In the search for the diverging time scale I consider two simpler models where I study intermittency and the mean waiting time in the two phases. The insights found are then applied to the Erdős-Rényi graph model. Gaining a thorough understanding of how the adaptive power method works and where it can still be improved lays the basis for applications on real world networks.

My thesis is divided into three chapters. Chapter 1 introduces the required theoretical background on graphs, Large Deviation Theory and the Adaptive Power Method algorithm. The numerical analysis of the algorithm is covered in chapter 2. Taking cue from observations in the analysis chapter that indicate a dynamical phase transition, I move onto studying this critical phenomenon in more detail in chapter 3.

Chapter 1

Theoretical Background

This chapter covers the definitions and concepts that build up to the central algorithm of my thesis, the *Adaptive Power Method*. I start with the definitions of a graph and a random walk. Further, the basics of Large Deviation Theory are introduced and the last sections will cover a special random walk, the *Driven Process*. Eventually, I state the algorithm of the *Adaptive Power Method*.

1.1 Graphs

A **graph**, also called a **network**, is often used to describe the structure of a system in a simplified way [1]. The connection between different parts of the system, called nodes, is represented by a link between them, called an edge.

Mathematically a graph is a tuple of two sets: **nodes** V and **edges** E. The nodes or vertices of a graph are elements in the countable vertex set V. Most often V is defined to be the integers $1, \dots, N$ under the assumption that it is finite with a total number of elements $N = |V|$. The second set E contains the edges where every edge is a tuple of two nodes, for example $(3, 8)$, connecting them. An edge is called undirected when the order of the two nodes does not matter, so $(3, 8)$ and $(8, 3)$ are the same edge. In short, the graph can be written as $G = (V, E)$. In this thesis I only deal with undirected and finite graphs, so $|V| < \infty$.

Random graphs are special graphs where the edges follow a random distribution. This means that one cannot speak about a specific graph with exactly given V and E but only of ensembles of graphs. In the ensemble the number of nodes is fixed to N, so $V = \{1, \dots, N\}$ and the procedure of constructing a specific E and thus the graph is specified. Let the total number of edges be denoted by $M = |E|$.

The **Erdős-Rényi (ER) random graph** can be constructed via two different ensembles [3]. I will use the *canonical* ensemble where N is fixed before hand and an edge between any two nodes is drawn with probability p. This results in an expected number of edges per node $\frac{\langle M \rangle_G}{N} = \frac{p(N-1)}{2}$, where $\langle M \rangle_G$ means the expectation value of M, the random variable for the number of edges, computed over the graph ensemble G. The probability distribution of generating a graph with $M = m$ edges is

$$\mathbb{P}(M = m) = \binom{\frac{N(N-1)}{2}}{m} p^m (1-p)^{\frac{N(N-1)}{2} - m}, \tag{1.1}$$

where $\frac{N(N-1)}{2}$ is the total number of possible edges [1]. $\mathbb{P}(M = m)$ is a binomial distribution so the probabilities of generating all individual graphs with m edges are the same.

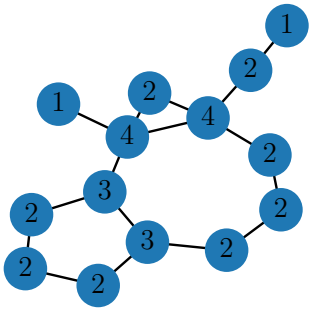


Figure 1.1: Largest connected component of an Erdős-Rényi random graph instance starting from $N = 15$ nodes and with $\bar{k} = 3$. The labels are the node degrees.

In comparison, the *micro-canonical* ensemble of the ER graph is defined by the set of all graphs with N nodes and m edges and selecting one graph out of that set with a uniform probability. In the large graph limit $N \to \infty$ with fixed $\frac{\langle M \rangle_G}{N}$ these ensembles become equivalent [3].

In a graph each node has a specific number of neighbors K, that is other nodes with which it shares an edge, called the **node degree**. The expected node degree of the ER graph ensemble is $\langle K \rangle_G = \bar{k} = p(N-1) = \frac{2\langle M \rangle_G}{N}$. In practice it is convenient to specify $\bar{k}$ and compute $p = \frac{\bar{k}}{N-1}$ when varying the graph size N because $\bar{k}$ should stay constant in the limit $N \to \infty$.

The **degree distribution** $\mathbb{P}(K = k)$ of the *canonical* ER graph ensemble, that is the probability that a node has degree $K = k$, is also binomial [1]. Each node can be connected with up to $N-1$ other nodes where an edge is created with success probability p, thus

$$\mathbb{P}(K = k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}. \tag{1.2}$$

In the limits $N \to \infty$ and $p \to 0$ but with fixed $\bar{k} = p(N-1)$, the degree distribution becomes a Poisson distribution, i. e.

$$\mathbb{P}(K = k) = \mathrm{e}^{-\bar{k}} \frac{(\bar{k})^k}{k!}. \tag{1.3}$$

Because the edges of the ER graph are created randomly it is important to know more about the connectivity of the graph. A **walk** on a graph is a sequence of nodes where the previous and next node have to be connected by an edge [3]. Through a walk one can reach two nodes, start and end, which do not necessarily have direct connection via an edge between them.

All nodes that can be reached through walks form a connected component of the graph. The ER random graph can have more than one component, which means that some nodes cannot be reached by a walk because they lie on another (disconnected) component. In the case of $\bar{k} > 1$ the graph will contain a **largest connected component** which includes almost all the nodes as $N \to \infty$ [15]. This largest connected component will be used equivalently for the ER random graph in all following sections.

Figure 1.1 shows the largest connected component of a realization of the ER graph with parameters $N = 15$ and $\bar{k} = 3$. The nodes (blue circles), are labeled with their node degree k and the lines connecting the nodes represent the edges. A special part of the graph shown is the so called **dangling chain**. This includes a node of degree 1 followed by one or more nodes of degree 2, seen in the upper right of the graph in figure 1.1. The nodes that are not in a dangling chain are referred to as the **bulk** of the graph.

1.2 Random Walks

A **random walk** is in general any process where the system is in one specific state at a time and can move from one state to another with a certain probability. This requires a state space V which can for example be the set of nodes of a graph. In the later sections, V will always be the set of nodes of the largest connected component of an ER graph, see figure 1.1 for an example.

The type of random walk that I consider in this thesis is a **discrete-time Markov chain** $X = (X_0, X_1, \ldots, X_{t-1})$ on a finite state space given by the graph G. The sequence X of t states needs to fulfill the **Markov property** which means that the probability of making a new step X_t in the sequence only depends on the last state of chain X_{t-1} [16]. Given a realization $x = (x_0, x_1, \ldots, x_{t-1})$ where $x_l \in V$ for all $l = 0, \ldots, t-1$, the Markov property means that the probability of the realization

$$\mathbb{P}(X = x) = \mathbb{P}(X_0 = x_0) \prod_{l=1}^{t-1} \mathbb{P}(X_l = x_l \mid X_{l-1} = x_{l-1}) \tag{1.4}$$

is factorized into one-step conditional probabilities $\mathbb{P}(X_l = x_l \mid X_{l-1} = x_{l-1}) =: \mathbb{P}(x_l \mid x_{l-1})$. The term *finite state space* for the chain comes from the definition of V, it requires that V is finite, and *discrete time* means that the sequence of states is indexed by the integer time $l = 0, \ldots, t-1$.

The **transition probability** $\mathbb{P}(j \mid i)$ is the probability to make a step to node j under the condition that the previous node was i. Since the state space V is finite thus indexed from 1 to N, the transition probabilities can be written as a matrix $\Pi(i, j) = \mathbb{P}(j \mid i)$. In this form Π is a row normalized stochastic matrix meaning $\Pi(i, j) \geq 0$ forall $i, j \in V$ and $\sum_{j \in V} \Pi(i, j) = 1$ for all $i \in V$.

The Markov chain is called **ergodic** if it is irreducible, that means each pair of states can be connected by a walk between them, and aperiodic, which means that the return time to a state is random [16]. If the state space comes from an ER graph, then ergodicity is achieved by choosing the largest connected component as the state space.

The **unbiased random walk** (URW) on a graph assigns equal probability to step to any of the neighbors of node i. Its transition matrix is

$$\Pi(i, j) = \frac{A(i, j)}{k_i}, \tag{1.5}$$

where the adjacency matrix element $A(i, j)$ is one if the nodes i and j are connected by an edge and zero otherwise. $k_i = \sum_{j \in V} A(i, j)$ is, as before, the degree of node i.

On the random walk trajectory X one can define **observables**. Given a function $f : V \longrightarrow \mathbb{R}$ that assigns a value to the nodes, the one-point additive observable

$$C_t = \frac{1}{t} \sum_{l=0}^{t-1} f(X_l) \tag{1.6}$$

is defined in terms of the chain X. C_t is the average of all node values mapped by f that the random walk encounters. For $f(i) = k_i$ mapping to the node degree, the resulting observable

$$C_t = \frac{1}{t} \sum_{l=0}^{t-1} k_{X_l} \tag{1.7}$$

is the mean visited degree of the random walk.

All the information to compute the one-point additive observables is contained in the **empirical occupation measure**

$$\rho_t(i) = \frac{1}{t}\sum_{l=0}^{t-1} \delta_{X_l,i} \quad \forall i \in V, \tag{1.8}$$

where $\delta_{a,b}$ is the Kronecker delta. This is a vector which counts the number of visits of each node normalized by the length t of the Markov chain. With the empirical occupation measure the observable is computed as

$$C_t = \sum_{i\in V} f(i)\rho_t(i). \tag{1.9}$$

For an ergodic Markov chain the empirical occupation measure converges in the limit $t \to \infty$ to the **stationary distribution** p. The stationary distribution is the solution to the eigenvalue problem $p\Pi = p$. This results in the ergodic theorem which states that the time average C_t converges to the ensemble average [17]

$$c^* = \lim_{t\to\infty} C_t = \sum_{i\in V} f(i)p(i), \tag{1.10}$$

where c^* is called the **typical event** or **ergodic value** of the observable C_t. The URW, whose transition matrix is equation (1.5), has the stationary distribution

$$p(i) = \frac{k_i}{2M} \quad \forall i \in V, \tag{1.11}$$

where $M = \frac{1}{2}\sum_{i\in V} k_i$ is the total number of edges [2].

1.3 Large Deviation Theory

The observable C_t is a function of the random walk X and thus a random variable. One can ask for the probability $\mathbb{P}(C_t = c)$ that the observable takes the value c. This probability distribution is concentrated around c^* and values c around it are called **fluctuations** of C_t.

The key idea of **large deviation theory** is that the distribution of the observable can be written as

$$\mathbb{P}(C_t = c) \asymp \mathrm{e}^{-tI(c)}, \tag{1.12}$$

where the function $I(c) \geq 0$ is called the **rate function** [5, 7]. The symbol $\asymp$ means that C_t follows the large deviation principle [5], which is that the limit

$$I(c) = \lim_{t\to\infty} -\frac{1}{t}\log \mathbb{P}(C_t = c) \tag{1.13}$$

exists and is finite. The larger the rate function, the less likely the fluctuation c. The most likely value is the minimum and zero of I which is the typical value c^* introduced in equation (1.10) [5].

The **Gärtner-Ellis theorem** [5] allows to compute the rate function indirectly. With the **scaled cumulant generating function** (SCGF)

$$\Psi(s) = \lim_{t\to\infty} \frac{1}{t}\log \mathbb{E}[\mathrm{e}^{tsC_t}] \tag{1.14}$$

the rate function I is the **Legendre-Fenchel transform**

$$I(c) = \sup\{sc - \Psi(s) \mid s \in \mathbb{R}\} \tag{1.15}$$

of Ψ if the latter is differentiable. The **moment generating function** (MGF) $G(s) = \mathbb{E}[\mathrm{e}^{tsC_t}]$ appearing in the SCGF can be computed in most cases easier than $\mathbb{P}(C_t = c)$ and so the step via the Gärtner-Ellis theorem from Ψ to I is the common one to take. The expectation $\mathbb{E}[\_]$ in the MGF G can be evaluated over the distribution of the realizations of the Markov chain $\mathbb{P}(X = x)$ or over the desired distribution $\mathbb{P}(C_t = c)$. Inserting the large deviation approximation (1.12) into equation (1.14) leads to a heuristic derivation of (1.15) [5].

The supremum can be simplified in the case of a differentiable Ψ, such that, given the unique solution $\tilde{s}$ to $c = \Psi'(\tilde{s})$, the rate function I is just the **Legendre transform**

$$I(c) = \tilde{s}c - \Psi(\tilde{s}) \tag{1.16}$$

of Ψ. The same relation appears in equilibrium statistical mechanics where the micro-canonical entropy is the rate function and its Legendre transform is the canonical free energy which is a SCGF function [5]. The inverse temperature has the role of the parameter s.

In the setting where the observable C_t is obtained from the random walk (section 1.2) the SCGF is determined by an **eigenvalue problem** as shown below. First compute the MGF that is the expectation over all possible trajectories:

$$G(s) = \mathbb{E}[\mathrm{e}^{tsC_t}] \tag{1.17}$$

$$= \sum_{x_0,\ldots,x_{t-1}=1}^{N} \mathbb{P}((X_0,\ldots,X_{t-1}) = (x_0,\ldots,x_{t-1}))\mathrm{e}^{tsC_t}, \tag{1.18}$$

where the $x_0,\ldots,x_{t-1}$ are summed over all possible states, which are the N nodes of the graph. Then use the Markov property in equation (1.4) to expand the probability of the realization

$$\mathbb{P}((X_0,\ldots,X_{t-1}) = (x_0,\ldots,x_{t-1})) = \mathbb{P}(X_0 = x_0)\prod_{l=1}^{t-1}\mathbb{P}(X_l = x_l \mid X_{l-1} = x_{l-1}) \tag{1.19}$$

$$= p(x_0)\prod_{l=1}^{t-1}\Pi(x_{l-1}, x_l), \tag{1.20}$$

where in the last line I abbreviate $p(x_0) = \mathbb{P}(X_0 = x_0)$ which is the distribution of the initial condition. Because the observable C_t is one-point additive, the exponential

$$\mathrm{e}^{tsC_t} = \mathrm{e}^{s\sum_{l=0}^{t-1} f(x_l)} \tag{1.21}$$

$$= \prod_{l=0}^{t-1}\mathrm{e}^{sf(x_l)} \tag{1.22}$$

can be written as a product as well. Define the **tilted matrix**

$$\tilde{\Pi}_s(i,j) = \mathrm{e}^{sf(i)}\Pi(i,j), \tag{1.23}$$

and insert equations (1.20) and (1.22) into equation (1.18) to simplify the MGF

$$G(s) = \sum_{x_0,\dots,x_{t-1}=1}^{N} p(x_0)\mathrm{e}^{sf(x_{t-1})} \prod_{l=1}^{t-1} \mathrm{e}^{sf(x_{l-1})}\Pi(x_{l-1},x_l) \tag{1.24}$$

$$= \sum_{x_0,\dots,x_{t-1}=1}^{N} p(x_0)\mathrm{e}^{sf(x_{t-1})} \prod_{l=1}^{t-1} \tilde{\Pi}_s(x_{l-1},x_l) \tag{1.25}$$

$$= \sum_{x_{t-1}=1}^{N} \mathrm{e}^{sf(x_{t-1})} \left(\sum_{x_{t-2}=1}^{N} \cdots \left(\sum_{x_1=1}^{N} \left(\sum_{x_0=1}^{N} p(x_0)\tilde{\Pi}_s(x_0,x_1) \right) \tilde{\Pi}_s(x_1,x_2) \right) \cdots \tilde{\Pi}_s(x_{t-2},x_{t-1}) \right) \tag{1.26}$$

$$= \sum_{x_{t-1}=1}^{N} \sum_{x_0=1}^{N} p(x_0)\tilde{\Pi}_s^{t-1}(x_0,x_{t-1})\mathrm{e}^{sf(x_{t-1})}. \tag{1.27}$$

In the last step the sums are grouped together into the repeated vector-matrix multiplications with $\tilde{\Pi}_s$. The expectation value is therefore the dot product of the vectors $p\tilde{\Pi}_s^{t-1}$ and $\mathrm{e}^{sf(-)}$.

For large t the repeated product of $\tilde{\Pi}_s$ is dominated by the left eigenvector l_s corresponding to the largest eigenvalue ζ_s of $\tilde{\Pi}_s$. ζ_s is largest in the sense that it has the largest absolute value of all existing eigenvalues of $\tilde{\Pi}_s$. Assuming p and l_s are normalized such that their elements sum to one, then $p\tilde{\Pi}_s^{t-1} \approx \zeta_s^{t-1} l_s$ and

$$G(s) \approx \zeta_s^{t-1} \sum_{x=1}^{N} l_s(x)\mathrm{e}^{sf(x)} \tag{1.28}$$

$$= \zeta_s^{t-1} a_s. \tag{1.29}$$

The constant $a_s = \sum_{x=1}^{N} l_s(x)\mathrm{e}^{sf(x)}$ does not depend on t, thus

$$\Psi(s) = \lim_{t\to\infty} \frac{1}{t} \log(G(s)) \tag{1.30}$$

$$= \lim_{t\to\infty} \frac{1}{t} \log(\zeta_s^{t-1} a_s) \tag{1.31}$$

$$= \lim_{t\to\infty} \frac{t-1}{t} \log(\zeta_s) + \frac{1}{t}\log(a_s) \tag{1.32}$$

$$= \log(\zeta_s). \tag{1.33}$$

The problem of computing the SCGF in the case of and discrete-time Markov chains on finite state spaces is reduced to finding the dominant eigenvalue ζ_s of the titled matrix $\tilde{\Pi}_s$ [6].

1.4 Driven Process

To simulate **rare events** c, which are values of C_t far off from c^* where $\mathbb{P}(C_t = c)$ is very small, one can use the **driven process** [7, 8]. This is a modified Markov process depending on the parameter s and whose typical event is a rare event of the unbiased random walk (URW). The asymptotic equivalence of the driven process with the URW conditioned on c is described in detail by Chetrite and Touchette [18].

The central idea is to bias the original Markov process, which yields the distribution $\mathbb{P}(C_t = c)$, to yield a new distribution

$$\mathbb{P}_s(C_t = c) = \frac{\mathrm{e}^{tsc}}{\mathbb{E}[\mathrm{e}^{tsc}]}\mathbb{P}(C_t = c) \tag{1.34}$$

$$\asymp \frac{\mathrm{e}^{tsc}}{\mathrm{e}^{-t\Psi(s)}}\mathrm{e}^{-tI(c)} \tag{1.35}$$

$$\asymp \mathrm{e}^{-tI_s(c)}, \tag{1.36}$$

where equations (1.12) and (1.14) were used and in the last step the terms in the exponent were grouped into a rate function for the driven process

$$I_s(c) = I(c) - (sc - \Psi(s)). \tag{1.37}$$

By the Gärtner-Ellis theorem (1.15) it follows that the specific $c = \Psi'(s)$ is a minimum and zero of $I_s(c)$. Thus the rare event c of the URW becomes a typical event of the driven process.

Chetrite and Touchette [18] state in Appendix E that the driven process has the following transition matrix

$$\Pi_s(i,j) = \frac{\tilde{\Pi}_s(i,j)r_s(j)}{\zeta_s r_s(i)}. \tag{1.38}$$

The tilted matrix $\tilde{\Pi}_s$ is defined in equation (1.23) and ζ_s is the dominant eigenvalue of $\tilde{\Pi}_s$ with the corresponding right eigenvector r_s. The Perron-Frobenius theorem for non-negative matrices ensures that $\zeta_s > 0$ exists and that all components of r_s are positive [19]. From the definition it follows that $\sum_{j=1}^{N} \Pi_s(i,j) = 1$ for all $i = 1, \ldots, N$.

The construction of the driven process is explained in appendix C of Chetrite and Touchette [20] and appendix E of Chetrite and Touchette [18]. The stationary distribution of Π_s is $\rho_s(i) = l_s(i)r_s(i)$ for all $i = 1, \ldots, N$ where l_s is the corresponding left eigenvector of $\tilde{\Pi}_s$.

The main difficulty in simulating the driven process is in finding the dominant eigenvalue and the corresponding right eigenvector of the tilted matrix. After that the transition matrix in equation (1.38) can be used to directly simulate trajectories. The next sections go into the details of solving the eigenvalue problem $\tilde{\Pi}_s r_s = \zeta_s r_s$.

1.5 Power Method

The **power method** is a relatively simple iteration scheme to compute the dominant eigenvalue and eigenvector of a matrix $M \in \mathbb{R}^{N\times N}$. Each iteration step l consists of the matrix-vector product $r^{(l+1)} = Mr^{(l)}$ which updates the estimated right eigenvector r. Assume that a basis of N eigenvectors v_i of M with eigenvalues λ_i sorted decreasingly $|\lambda_1| > |\lambda_1| > \cdots > |\lambda_N|$ exist. Then the initial condition

$$r_s^{(0)} = \sum_{i=1}^{N} c_i v_i \tag{1.39}$$

is a linear combination of the eigenvectors and the coefficients c_i are uniquely determined by the initial condition. The repeated matrix-vector product results in

$$r_s^{(l)} = M^l r^{(0)} \tag{1.40}$$

$$= \sum_{i=1}^{N} c_i M^l v_i \tag{1.41}$$

$$= \sum_{i=1}^{N} c_i \lambda_i^l v_i \tag{1.42}$$

$$= \lambda_1^l \left(c_1 v_1 + \sum_{i=2}^{N} c_i \left(\frac{\lambda_i}{\lambda_1} \right)^l v_i \right). \tag{1.43}$$

Because λ_1 is the dominant eigenvalue, i. e. it has the largest magnitude, all factors obey to

$$\left| \frac{\lambda_i}{\lambda_1} \right| < 1 \quad \forall i = 2, \ldots, N. \tag{1.44}$$

So in the limit

$$\lim_{l \to \infty} \left(\frac{\lambda_i}{\lambda_1} \right)^l = 0 \quad \forall i = 2, \ldots, N \tag{1.45}$$

and $r^{(l)}$ converges to v_1. The speed of the convergence depends on the eigenvalue gap $|\lambda_1| - |\lambda_2|$ as the factor $\frac{\lambda_2}{\lambda_1}$ is the largest one and thus dominantes the sum in equation (1.43).

After each step l the estimate of λ_1 is given by the ratio

$$\frac{r^{(l+1)}(i_0)}{r^{(l)}(i_0)} \tag{1.46}$$

at some fixed index $i_0 \in \{1, \ldots, N\}$. For numerical stability it is better to rescale the vector $r^{(l+1)}$ to avoid its norm from diverging or going to zero.

1.6 Adaptive Power Method

In this section, I will introduce the **Adaptive Power Method** (APM) to compute the dominant eigenvalue and eigenvector of the tilted matrix $\tilde{\Pi}_s$. First, the algorithm is stated and then explanations of individual steps and an extension follow.

The APM simulates a random walk where it starts as the unbiased random walk (URW) and finishes as the driven process [10]. During the random walk the eigenvalue ζ_s and the eigenvector r_s are learned such that the final transition matrix is the one of the driven process in equation (1.38). The variables of the driven process are labeled with a superscript $^{(l)}$ to indicate the value at the time step l.

The **initialization** of the algorithm is done by starting with the transition matrix of the URW in equation (1.5). The unbiased dominant eigenvalue is $\zeta_0 = 1$ with an unbiased right eigenvector of all ones, since Π is a row normalized stochastic matrix. The value of s is selected and kept fixed.

The initial values at $l = 0$ are

$$\Pi_s^{(0)} = \Pi, \tag{1.47}$$
$$r_s^{(0)} = (1, \ldots, 1)^T, \tag{1.48}$$
$$\zeta_s^{(0)} = 1, \tag{1.49}$$
$$\gamma^{(0)} = (1, \ldots, 1)^T, \tag{1.50}$$
$$a_0 = 1, \tag{1.51}$$
$$x_0 \text{ initial node}, \tag{1.52}$$
$$C_s^{(0)} = f(x_0). \tag{1.53}$$

During each step $l = 1, \ldots, t-1$ do the following:

1. Use the label $i = x_{l-1}$ for the previously visited node.
2. Select a new node $j = x_l$ with the probabilities given by the row vector $\Pi_s^{(l-1)}(i, \_)$.
3. Save the observable increments

$$C_s^{(l)} = f(j) \quad \text{and} \quad K_s^{(l)} = -\log\left(\frac{\Pi(i,j)}{\Pi_s^{(l-1)}(i,j)}\right). \tag{1.54}$$

4. Update the i-th component of the right eigenvector

$$r_s^{(l)}(i) = r^{(l-1)}(i) + a_{l-1}\left(\frac{\mathrm{e}^{sf(i)}\gamma^{(l-1)}(i)}{\zeta_s^{(l-1)}} - r^{(l-1)}(i)\right). \tag{1.55}$$

5. Update the eigenvalue

$$\zeta_s^{(l)} = \max\{r_s^{(l)}(m) : m = 1, \ldots, N\}. \tag{1.56}$$

6. Update the normalization factor

$$\gamma^{(l)}(i) = \sum_{m=1}^{N} \Pi(i,m) r_s^{(l)}(m). \tag{1.57}$$

7. Update the i-th row in the transition matrix

$$\Pi_s^{(l)}(i,m) = \frac{\Pi(i,m) r_s^{(l)}(m)}{\gamma^{(l)}(i)} \quad \text{for all } m = 1, \ldots, N. \tag{1.58}$$

8. Update the learning rate $a_l = l^{-\alpha}$

For a long enough time t each value at the final step $l = t-1$ converges to the one of the driven process in equation (1.38). The approximate values are

$$\Pi_s \approx \Pi_s^{(t-1)}, \tag{1.59}$$
$$r_s \approx r_s^{(t-1)}, \tag{1.60}$$
$$\zeta_s \approx \zeta_s^{(t-1)}, \tag{1.61}$$
$$\Psi(s) \approx \Psi_s := \log(\zeta_s^{(t-1)}), \tag{1.62}$$
$$c^* \approx C_{t,s} := \frac{1}{t}\sum_{l=0}^{t-1} C_s^{(l)}, \tag{1.63}$$
$$I(c^*) \approx K_{t,s} := \frac{1}{t-1}\sum_{l=1}^{t} K_s^{(l)}, \tag{1.64}$$
$$\Psi(s) \approx \Psi_{t,s} := sC_t - K_t. \tag{1.65}$$

The value c^* is the typical value of the driven process and the solution to the equation $s = I'(c^*)$ or $c^* = \Psi'(s)$, where the parameter s is the one selected in the beginning.

1.6.1 Explanation of the APM

The form of the transition matrix of the driven process, equation (1.38), can be simplified to

$$\Pi_s(i,j) = \frac{\Pi(i,j) r_s(j)}{\gamma(i)} \tag{1.66}$$

where $\gamma(i) = \zeta_s r_s(i) \mathrm{e}^{-sf(i)}$ is the row normalization factor.

The iterative update of the right eigenvector r_s, equation (1.55), is a modified version of the power method, see section 1.5. Below I start from the matrix-vector product $\tilde{\Pi}_s r_s$ and rewrite the i-th component of that product:

$$r_s^{(l)}(i) = (\tilde{\Pi}_s r_s^{(l-1)})(i) \tag{1.67}$$
$$= \sum_{j=1}^{N} \tilde{\Pi}_s(i,j) r_s^{(l-1)}(j) \tag{1.68}$$
$$= \sum_{j=1}^{N} \mathrm{e}^{sf(i)} \Pi(i,j) r_s^{(l-1)}(j) \tag{1.69}$$
$$= \sum_{j=1}^{N} \mathrm{e}^{sf(i)} \frac{\Pi(i,j)}{\Pi_s^{(l-1)}(i,j)} r_s^{(l-1)}(j)\, \Pi_s^{(l-1)}(i,j) \tag{1.70}$$
$$= \mathrm{e}^{sf(i)} \mathbb{E}_{\Pi_s^{(l-1)}}[R_s^{(l-1)}(i,j) r_s^{(l-1)}(j)]. \tag{1.71}$$

In the last step I introduce the likelihood ratio

$$R_s^{(l-1)}(i,j) = \frac{\Pi(i,j)}{\Pi_s^{(l-1)}(i,j)} = \frac{\gamma^{(l-1)}(i)}{r_s^{(l-1)}(j)} \tag{1.72}$$

and the sum over j with the weighting factor $\Pi_s^{(l-1)}(i,j)$ is written as an expectation value with this transition matrix $\Pi_s^{(l-1)}$, which is the transition matrix of the APM at time step l.

The argument inside the expectation simplifies to

$$R_s^{(l-1)}(i,j)r_s^{(l-1)}(j) = \frac{\gamma^{(l-1)}(i)r_s^{(l-1)}(j)}{r_s^{(l-1)}(j)} \tag{1.73}$$

$$= \gamma^{(l-1)}(i). \tag{1.74}$$

Putting things together this updates the i-th component of the right eigenvector after one step of the Markov chain by the rule

$$r_s^{(l)}(i) = \mathrm{e}^{sf(i)}\gamma^{(l-1)}(i). \tag{1.75}$$

As $i = x_{l-1}$, only one component of r_s gets updated. For most vector norms this would require a renormalization. A solution is to use the **infinity** or **maximum norm** where $\|r_s\|_\infty = \max\{|r_s(m)| : m = 1,\dots,N\}$ and fixing the norm to be $\zeta_s = \|r_s\|_\infty$. Then the updated component or r_s only needs to be divided by the eigenvalue ζ_s as the matrix-vector product scales the vector by the eigenvalue, see equation (1.46). The update rule

$$r_s^{(l)}(i) = \frac{\mathrm{e}^{sf(i)}\gamma^{(l-1)}(i)}{\zeta_s^{(l-1)}} \tag{1.76}$$

automatically normalizes r_s in the maximum norm to ζ_s.

The last modification that is missing to get to equation (1.55) is the introduction of a **learning rate** $a_l \in (0,1]$. This is needed because the change

$$\Delta r_s^{(l)}(i) = \frac{\mathrm{e}^{sf(i)}\gamma^{(l-1)}(i)}{\zeta_s^{(l-1)}} - r_s^{(l-1)}(i) \tag{1.77}$$

between the update in equation (1.76) and the previous value is stochastic. Even as the APM converges one expects fluctuations in $\Delta r_s^{(l)}(i)$ which are noisy around zero, these need to be counter balanced [10]. The update rule in equation (1.55) adds the stochastic update from equation (1.77) scaled by the learning rate a_l to the i-th component of r_s. In order to avoid that the noise accumulates and breaks the convergence, a_l is decreased towards zero for large l.

The update of the normalization factor $\gamma^{(l)}$ in step 6 and the update of the transition matrix $\Pi_s^{(l)}$ in step 7 follow because a component of r_s appearing in them is changed. The eigenvalue is updated in step 5 by taking the maximum value of the components of r_s because the normalization of r_s is chosen to be ζ_s in the maximum norm.

The APM simulates a Markov chain whose transition matrix converges to the one of the driven process. Thus the additive observable $C_{t,s}$ in equation (1.63) converges to the typical value of the driven process c^*. This is the rare event that I want to simulate and which follows from the solution to $c^* = \Psi'(s)$ or $s = I'(c^*)$, see the Gärtner-Ellis theorem. That the observable $K_{t,s}$ from equation (1.64) converges to the value of the rate function $I(c^*)$ is a result of the special form of the driven process. For a derivation, see section 2 of Carugno et al. [21]. Given the two observables $C_{t,s}$ and $K_{t,s}$, the second estimate of $\Psi(s)$ is the empirical Legendre-transform $\Psi_{t,s}$ from equation (1.65).

In the end the APM allows to compute the value of the SCGF $\Psi(s)$ in two ways. The first is $\Psi_s = \log(\zeta_s)$ via the logarithm of the estimated dominant eigenvalue ζ_s while the second one is $\Psi_{t,s} = sC_{t,s} - K_{t,s}$ via the above mentioned additive observables $C_{t,s}$ and $K_{t,s}$.

1.6.2 Transfer Learning

A convergence speed-up of the APM is achieved through a **transfer learning** scheme [10]. In this scheme the final eigenvector r_s and transition matrix Π_s are used as the initial conditions to learn the new eigenvector $r_{s+\Delta s}$ and transition matrix $\Pi_{s+\Delta s}$ where the s-parameter is changed only by a small amount Δs. The number of iterations needed to learn the updated eigenvector is then reduced because the difference between the initial r_s and the final $r_{s+\Delta s}$ is small for small Δs. The transfer learning procedure can be repeated by starting at $s_0 = 0$ and arriving at a final s_1 after $\frac{s_1 - s_0}{\Delta s}$ epochs, which are the time step segments of fixed s. The learning rate a_l needs to be reset after each epoch.

Chapter 2

Numerical Analysis

The algorithm of the Adaptive Power Method (APM) was described in the previous chapter. This chapter covers the numerical analysis of the method. First, I look at the convergence time of the APM and how the transfer learning scheme improves it. I also study how the SCGF and the rate function that are computed via the APM compare to the exact forms. In the second half, I go over the effect of the learning rate and compare the APM with the power method. In the end, I briefly show how the APM could be extended to condition on a fixed value of the observable $C_t = c$, rather than fixing the tilting parameter s.

2.1 Convergence of the APM

During each step l of the APM the values of $\Psi_s^{(l)}$, $C_{l,s}$, $K_{l,s}$ and $\Psi_{l,s}$ are saved. They are now analyzed as *time series* of discrete time l. In order to have a reference *exact* value for the dominant eigenvalue ζ_s and thus $\Psi(s)$, I used Python's linear algebra eigenvalue solver to solve for the dominant eigenvalue and eigenvector directly.

I first look at the SCGF Ψ_s computed as the log of the dominant eigenvalue ζ_s of the tilted matrix in equation (1.23). Figure 2.1 shows the time series of Ψ_1, left plot (a), and Ψ_{-1}, right plot (b), for $t = 10^6$ steps. The solid lines are obtained by averaging over 100 repeated simulations and the shaded area represents the standard deviation. For reference the exact value of $\Psi(s)$ is indicated by the dashed horizontal lines. Plotted in different colors are the curves for four different graph sizes from 50 to 400 nodes increasing by factors of two.

The initial condition $\zeta_s^{(0)} = 1$ translates to $\Psi_s^{(0)} = 0$, which is outside the view in plot (a) because of the semi-logarithmic scale. For $s = 1$ the SCGF is positive and larger than 1 as seen in figure 2.1 (a) so the convergence happens from below. The number of steps needed to converge to the exact value $\Psi(s)$ increases, though not very drastically, with the graph size N as one can see from the wider error bands.

In the case $s = -1$ the convergence is much slower than for $s = 1$. A delay is visible in figure 2.1 (b) where Ψ_{-1} stays at zero for an initial period. This behavior is due to the initial conditions of ζ_s and r_s in equations (1.48) and (1.49). A negative SCGF, as for Ψ_{-1}, corresponds to $\zeta_s \in (0, 1)$. Because the maximum component of r_s is normalized to be ζ_s and initially all components of r_s have the value 1, the APM first has to visit all nodes of the graph before ζ_s is updated for the first time. This is visible in plot (b) where the period of $\Psi_{-1}^{(l)} = 0$ increases with the graph size N. After this initial period the averaged SCGF drops quickly and even falls below the exact value before converging to $\Psi(-1)$.

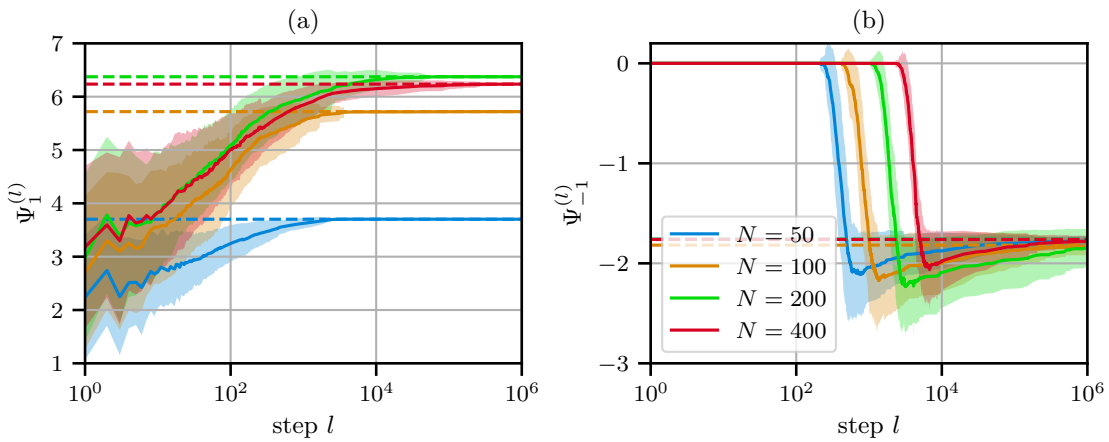


Figure 2.1: Averaged time series of the SCGF $\Psi_s^{(l)} = \log(\zeta_s^{(l)})$ from the APM. The value of s is changed from 1 to -1 between the left (a) and right (b) plot. The dashed line shows the exact value and the shaded area the standard deviation computed over 100 repetitions of the APM. The different color curves correspond to the different graphs sizes N. The fixed parameters are $\bar{k} = 3$ and $\alpha = 0.1$.

The parameters $\bar{k} = 3$ and $\alpha = 0.1$ used in these simulations of the APM are the same throughout the rest of the chapter.

2.1.1 Transfer Learning

The convergence of the APM improves when transfer learning is used. Especially for $s < 0$ the effect becomes important. Transfer learning, described in section 1.6.2, means that the already learned information of the right eigenvector r_s and the transition matrix Π_s for a previous value of s is used as the starting point of the APM with new value $s + \Delta s$. The APM is thus divided in epochs of different s values.

The time series of $\Psi_s^{(l)}$ with transfer learning is shown in figure 2.2. The APM is run for 5 epochs and $10\,000$ steps per epoch, thus in total only $50\,000$ steps compared to the 10^6 steps before. The initial epoch starts at $s = 0$ and the change between epochs is $\Delta s = 0.25$.

In the left plot (a) of figure 2.2 the time series converges to $\Psi(1)$ in a very clear step function shape. The error is reduced significantly and in all epochs. The step function shape indicates that the APM converges in each epoch to the temporary value $\Psi_s^{(l)}$.

When looking at the right plot (b) in figure 2.2 the first two epochs of $\Psi_{-1}^{(l)}$, that is up to step $l = 20\,000$, resemble very much what is shown in figure 2.1. The second epoch, where $s = -0.25$, absorbs the time that is needed for the APM to first change $\Psi_{-1}^{(l)}$ from zero, thus in the later epochs this period is not present. One can see that the convergence time to the temporary values of $\Psi_{-1}^{(l)}$ is decreased and also the error is reduced. The time series become a similar step function as for the case $s = 1$.

The benefit of the transfer learning and the step function shape is that the value $\Psi(s)$ can be extracted for intermediate values of s. The time per epoch influences the accuracy as the APM needs to converge well enough in each epoch. Its value needs to be chosen according to the graph size N and increment Δs. A similar behavior as described above for $\Psi^{(l)}$ is observed for the estimator $\Psi_{t,s}$ whose plots are omitted to avoid redundancy.

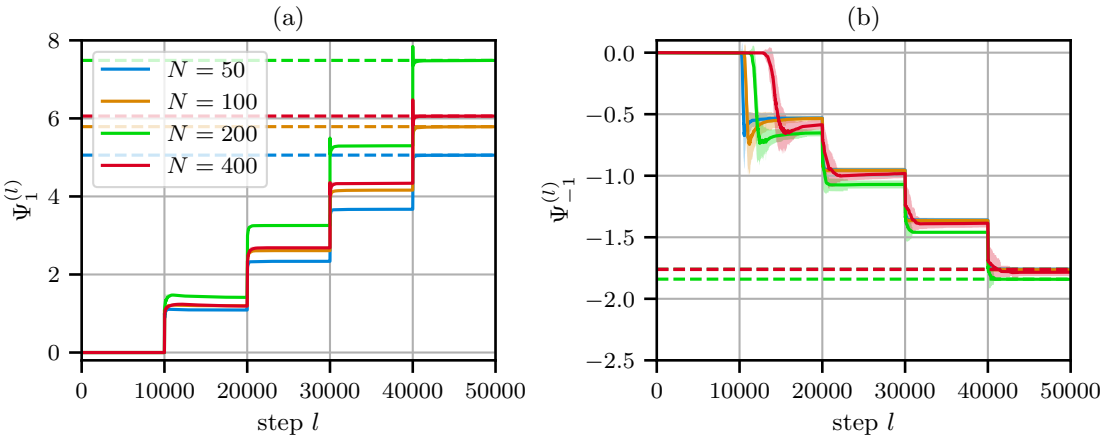


Figure 2.2: Averaged time series of the SCGF $\Psi_s^{(l)} = \log(\zeta_s^{(l)})$ from the APM with transfer learning. The time series consist of 5 epochs with 5000 steps each. The solid line are averages over 100 repetitions and the shaded area the standard deviation thereof. The different colors correspond to the different graph sizes N and the dashed lines show the exact value of $\Psi(s)$. The parameters used in the APM are $\bar{k} = 3$, $\alpha = 0.1$ and $\Delta s = 0.25$.

2.1.2 Rate Function

The pairs $(C_{t,s}, K_{t,s})$ computed with the transfer learning of the APM represent the rate function (RF) of the unbiased random walk. By the point-wise empirical Legendre transform $\Psi_{t,s} = sC_{t,s} - K_{t,s}$ one gets the SCGF as the pairs $(s, \Psi_{t,s})$. Both functions are shown in figure 2.3 for a graph of size $N = 400$ with different times t per epoch ranging from 500 to 4000. The data points represent the average of 100 simulations where the error bars represent the standard deviation.

The left plot (a) shows the SCGF whose exact curve was computed by finding the exact eigenvalue of the titled matrix and taking the logarithm of it. Notice that the error bars are only along the y-axis because the parameter s is kept fixed during each epoch of the APM and thus does not contain an error. The Legendre transform of the SCGF is the rate function which is shown in the right plot (b) of figure 2.3. The exact curve was computed as the numerical Legendre-Transform of the exact SCGF. Here, the data points obtained from the APM have an error along both the x- and the y-axis because both quantities $C_{t,s}$ and $K_{t,s}$ are observables that are averaged over 100 simulations of the APM.

The APM results of the SCGF align very well with the exact function for $s > 0$ for all different times t per epoch. For $s < 0$ the data points lie below the exact SCGF, which is due to the Legendre transform and that the data points of the rate function lie above the exact rate function. When doubling the time t the SCGF gets closer to the exact function also for $s < 0$. In the region of small negative s the SCGF seems to develop a kink, that is its slope changes rapidly.

That the estimates of the rate function shown in figure 2.3 (b) lie always above the exact curve (black line) is expected. Whenever a sampling method is used, implicit constraints are added into the larger system where the rate function I, in terms of the observable C_t, is an infimum of a higher dimensional rate function. See Touchette [5] for more on the infimum optimization process called contraction. These constraints on the infimum lead to an over-estimation of the rate function I by the APM [20].

Similar to the SCGF, the rate function obtained by the APM is very close to the exact one when $c > c^* \approx 4$ for all values of t plotted. When $c < c^*$ the accuracy depends strongly on the time t per epoch. For the shortest time span per epoch $t = 500$ (blue dots) the rate function shows a clear

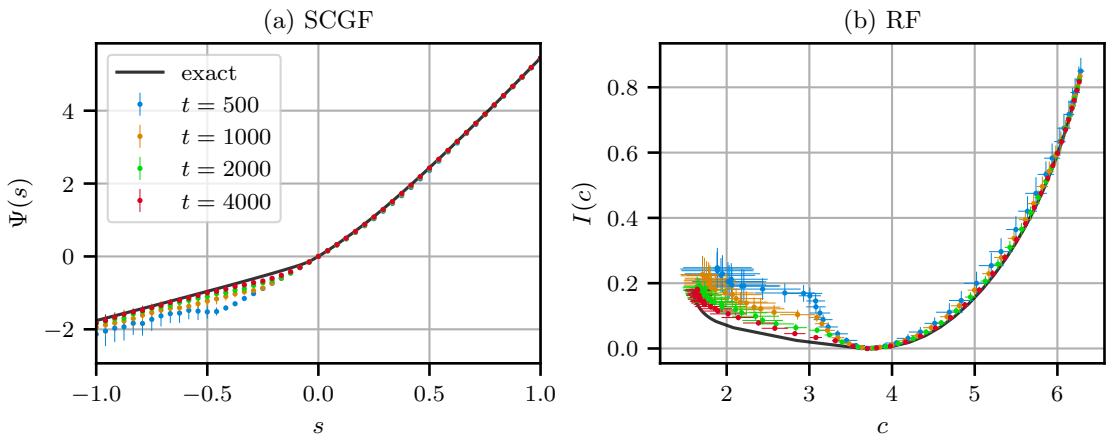


Figure 2.3: Scaled cumulant generating function (SCGF) $\Psi(s)$ and rate function (RF) $I(c)$ computed from the APM with transfer learning. The graph size is $N = 400$ with $\bar{k} = 3$. The number of iterations per epoch t is increased by factors of 2. Each data point is an average over 100 repetitions of the APM and the error bars are the standard deviations. The APM SCGF is $(s, sC_t - K_t)$ and the rate function (C_t, K_t), where the exact SCGF is the logarithm of the exactly computed eigenvalue of the titled matrix and the exact rate function is the numerical Legendre transform of the exact SCGF. The APM is run with $\alpha = 0.1$.

dip. In this case the APM does not have enough time to visit the whole graph during the first epochs, recall figure 2.1 (b). This means that the estimators $C_{t,s}$ and $K_{t,s}$ actually do not sample the true values during those epochs. For longer times t the accuracy of the APM increases and the almost linear section of the rate function is recovered. Nevertheless, the difference to the exact I remains the largest along the linear stretch, which is by the Legendre-Fenchel transform related to the kink of the SCGF.

The limits of s for the SCGF Ψ in figure 2.3 (a) are fixed to be exactly $s = \pm 1$ while the limits for c in the rate function are determined by the asymptotic slopes of Ψ. These slopes represent the minimally and maximally visited averaged degrees on the graph. If there exists a dangling chain then the minimum is always 1.5 as the random walk spends most of its time on the end of the chain (degrees 1 and 2). In contrast, the maximum depends on two neighboring nodes of largest degrees.

The behavior can be seen in the stationary distribution of the driven process which is shown on a graph representation in figure 2.4. The colors describe the value of the stationary distribution

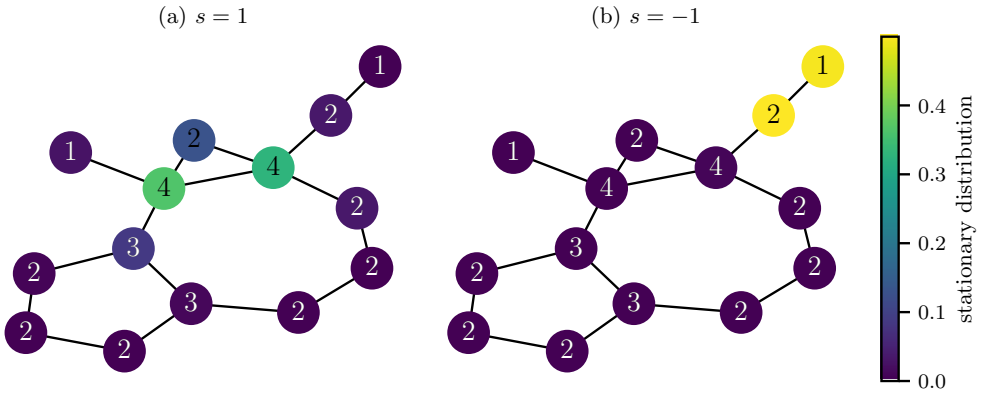


Figure 2.4: Stationary distribution of the driven process for (a) $s = 1$ and (b) $s = -1$. The graph is the largest connected component of an ER graph with $N = 15$ and $\bar{k} = 3$. Node labels are the degrees of the nodes.

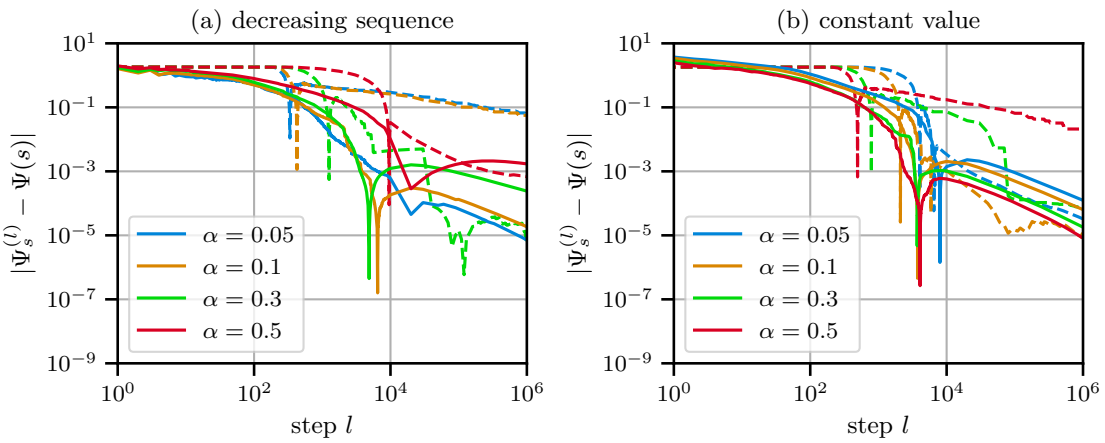


Figure 2.5: Time series of the averaged absolute error of the SCGF during the APM. The form of the learning rate is varied between (a) with a decreasing learning rate $a_l = l^{-\alpha}$ and (b) with a constant learning rate $a_l = \alpha$. The curves for different values of α are shown. The solid lines correspond to $s = 1$ and the dashed ones to $s = -1$ simulations. Averaging is done over 100 trajectories with a graph of size $N = 50$ and $\bar{k} = 3$.

on each node which highlights on which nodes the random walk spends most of the time. In the case (a) it is $s = 1$, and the nodes with the highest degrees 4 are visited the most while in the other case (b), where $s = -1$, the nodes in the dangling chain are visited (degree 1 and 2). The yellow color of the chain nodes in (b) shows that most of the stationary distribution weight is in the chain meaning that the random walk is localized there. In (a) the weight is spread out over more nodes in the bulk (green and blue color) indicating that the random walk is de-localized.

2.2 Learning Rate

The learning rate $a_l = l^{-\alpha}$ is a decreasing sequence with time l. If one is unfamiliar with stochastic approximation this might seem a bit unintuitive as the effect of the update weighted by a_l is decreased. It is important to keep in mind that the update is noisy and only on average steers in the right direction, so the learning rate influences how strong the effect of the noise is. Under certain circumstances a constant learning rate $a_l = \alpha$ for all l could give still acceptable results if α is chosen small enough [22]. In this section I investigate the effect of a constant learning rate on the APM with a small graph size of $N = 50$ with $\bar{k} = 3$.

The absolute error $|\Psi_s - \Psi(s)|$ is shown in figure 2.5. The curves are based on an averaged Ψ_s over 100 trajectories of length $t = 10^6$. The case $s = 1$ is plotted in solid lines while $s = -1$ are the dashed lines. Colors indicate the different values of the learning rate parameter α. Surprisingly, in comparison between the left (a) where the learning rate is the decreasing sequence and the right (b) with the constant learning rate the error is not significantly different. Similarly, the plots of the rate function ($t = 2000$) for both cases of the learning rate, shown in figure 2.6, do not show a significant difference. One can only see that the value $\alpha = 0.1$ in the decreasing learning rate case yields very good results while for the constant learning rate this would be for $\alpha = 0.5$.

While this does not allow to make a general conclusion, it shows that for small graph sizes $N = 50$ and the topology of the ER graph the noise in the update of the dominant eigenvector r_s components seems to be not too large. Thus the constant learning rate performs comparably well as the decreasing learning rate sequence and it explains why the relatively slowly decaying sequence for $\alpha = 0.1$ is optimal.

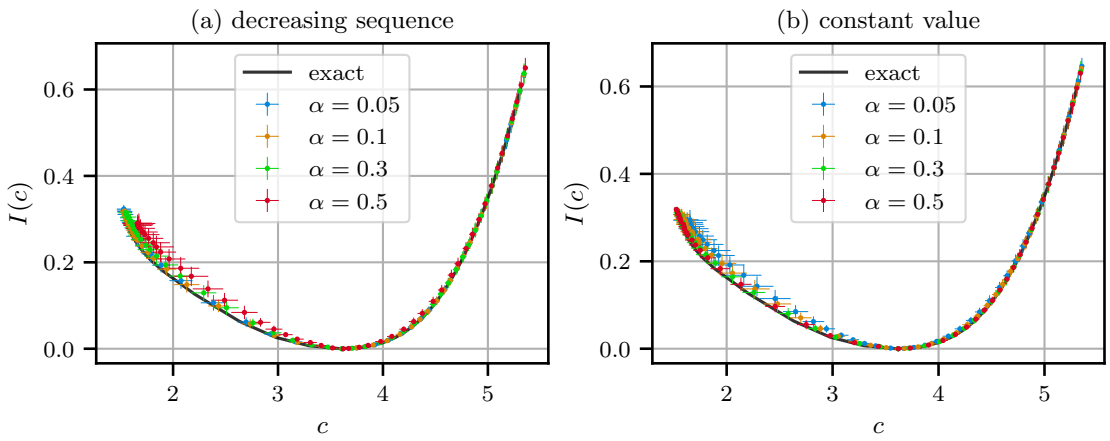


Figure 2.6: Rate function $I(c)$ computed from the APM through transfer learning with $t = 2000$ steps per epoch and in total 50 epochs. The learning rate is changed from (a), decreasing learning rate $a_l = l^{-\alpha}$, to (b), constant learning rate $a_l = \alpha$. The colors show the results for different values of α. The APM is repeated 100 times to compute the average and the standard deviation which is shown by the error bars.

2.3 Comparison with the Power Method

The power method (PM) described in section 1.5 is a well known method for solving for the dominant eigenvalue of a matrix. I compare the convergence of the power method with the APM for different graph sizes. The absolute error of the SCGF $\Psi(s) = \log(\zeta_s)$ computed with these two methods is shown in figure 2.7. The solid lines correspond to $s = 1$ and the dashed ones to $s = -1$.

In plot (a) of figure 2.7 the APM with transfer learning is simulated over 10 epochs with 1000 steps per epoch. The error decreases significant only in the last epoch which is due to the transfer learning which just reduces the total number of steps needed. The final error after $10\,000$ steps is in the order 10^{-1} to 10^{-3} and is lower for smaller graph sizes N.

The power method shown in (b) converges much faster and can reach even machine precision with an error below 10^{-14}. One can see that for larger graph sizes N and $s = -1$ the convergence becomes slower. An example is the dashed green curve for $N = 200$ which only decreases to an error of 10^{-9} after 1000 iterations. In order for the power method to converge, I need to use a learning rate as seen in the APM. A constant $a_l = \alpha = 0.1$ is sufficient because of the deterministic nature of the method. Note that the value of α influences the speed of convergence.

Overall the power method converges faster and to lower errors than the APM. This is expected since it updates the whole eigenvector r_s during each iteration while the APM changes only one component per step. The strength of the APM lies in that it simulates a random walk so the matrix Π_s does not need to be present in its full form. Only the local information of the neighbors of a node i, that is $\Pi_s(i, \_)$, is needed. This is specifically an advantage if one has no prior knowledge of the network and thus the full transition matrix or when the latter one is time dependent. See Di Bona et al. [13] for a maximally spreading APM in unknown networks.

Another advantage of the APM is the transfer learning which allows to compute the dominant eigenvalue for a range of s values in one simulation and update Π_s along the way. For the power method where s has to be fixed the whole time this is not possible, and one would need to compute the eigenvalues for each s separately which involved constructing $\tilde{\Pi}_s$ repeatedly.

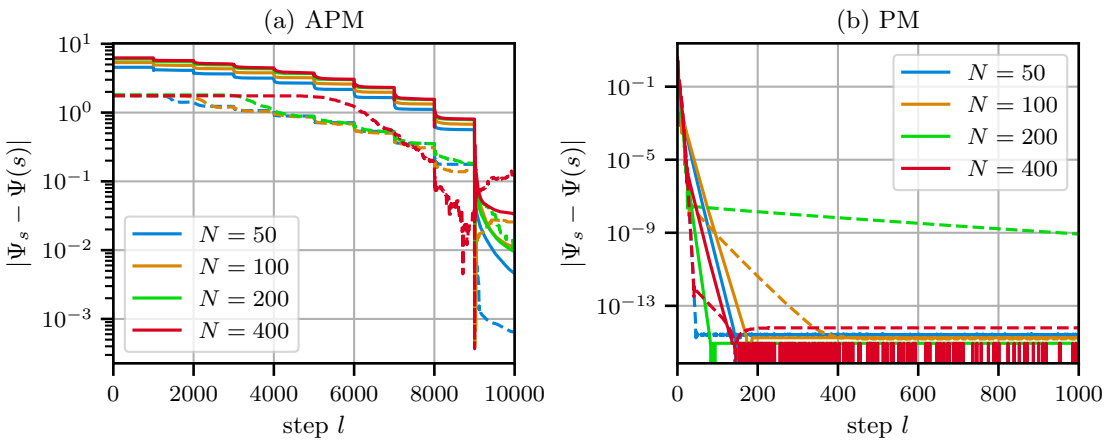


Figure 2.7: Absolute error of the SCGF computed (a) with the APM with transfer learning and (b) with the power method (PM). The solid lines correspond to $s = 1$ and the dashed lines to $s = -1$. The graph size N is doubled between the simulations. The fixed parameters where $\bar{k} = 3$ and $\alpha = 0.1$ for both methods. The APM curve is computed as the average over 100 repeated simulation.

2.4 Learning the s-parameter

The APM is introduced in section 1.6 with the parameter s chosen beforehand and controlling the algorithm. The value of the observable c that is realized by the APM is related to s via the derivative of the SCGF $\Psi'(s) = c$. But this requires the SCGF Ψ to be known and in a format where the derivative can be evaluated. For larger graphs computing the SCGF can be computationally too expensive. I show a modified transfer learning of the APM that can be used to simulate a trajectory for a desired value of c. This concept was initially introduced in the APM by Borkar et al. [9].

The idea is to fix a value of c and let the value of s be learned. This is done by adding another stochastic approximation scheme. A simple updating scheme which works as a proof of concept is

$$s^{(e+1)} = s^{(e)} - b_e(C_{t,s} - c), \tag{2.1}$$

where b_e is another learning rate and e the index of the epoch. During each epoch e the APM is run with a fixed $s^{(e)}$ and a value of $C_{t,s}$ is obtained. The error $C_{t,s} - c$ then determines the update of s. Similar to the previous learning rate, see step 8 of the APM, the b_e is a decreasing series learning rate

$$b_e = b_0 e^{-\beta} \tag{2.2}$$

with the exponent β and a scaling constant b_0 as parameters. The speed with which b_e decreases is slower than the one of a_l because the former is only changed in between epochs while the latter changes after every step. The value of b_0 influences mainly the first update of s.

An example of learning the s value for three different values $c = 2, 3$ and 4 is shown in figure 2.8. The left plot (a) shows the time series of $C_{t,s}$ and the right plot (b) shows the value of s at that time during the simulation. The epoch length is 6000 iterations and the graph size is $N = 50$ with $\bar{k} = 3$. The parameters for the learning rate in equation (2.2) are $b_0 = 0.1$ and $\beta = 1.2$, whose values were found by trial and error.

The plots show that the convergence of $C_{t,s}$ is not perfect which is expected since it is a random variable and thus shows fluctuations. But the general magnitude of s and its sign were

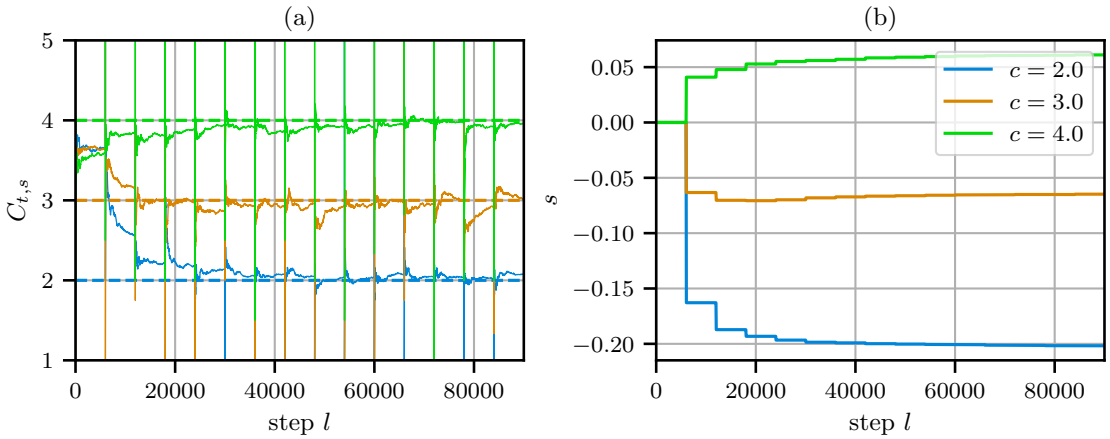


Figure 2.8: Simulations of the transfer learning APM to learn the parameter s on a graph with $N = 50$ and $\bar{k} = 3$. Three values $c = 2$, 3 and 4 are used. The left plot (a) shows the estimate $C_{t,s}$ over all epochs. The corresponding values of s in every epoch are plotted in (b).

learned, see right hand side (b) of figure 2.8. Values of c that are larger than the typical value c^* correspond to $s > 0$ while values of c that are less than the typical value correspond to $s < 0$. With further improvements this opens the usage of the APM to applications where the SCGF of a system is unknown or incomputable and conditioning on a fluctuation c is desired.

Chapter 3

Dynamical Phase Transition

In this chapter I investigate closer how the kink in the SCGF related to the driven process on the ER graph develops, which is believed to indicate a **dynamical phase transition** (DPT) . By large deviation theory the value of the observable C_t which is realized by the driven process is the fluctuation $c = \Psi'(s)$. Thus if the SCGF Ψ has a discontinuity in the derivative this means that the observable cannot be sampled at that point by the driven process.

The SCGF previously seen in figure 2.3 has two sections where it becomes asymptotically linear. These are the cases where $s \to \pm\infty$. In between, the derivative of the SCGF, which is the value c of the observable as a function of s, transitions from one asymptotic value to another. This transition becomes steeper as the graph size N increases (compare figure 3.12). It is thought that in the limit $N \to \infty$ this becomes a discontinuity and so the SCGF develops a kink. By the Legendre-Fenchel transformation this kink transforms into a linear section in the rate function (compare figure 2.3).

The values of c to the left and right of the discontinuity describe the difference dynamical phases. The typical trajectories that realize those fluctuations show a very different behavior. The idea was already briefly touched in the previous chapter where I describe the rate function. There figure 2.4 shows the stationary distribution on a small ER graph from which it becomes clear that the trajectories to the right of the critical point are de-localized in the bulk and have a value of $c \approx c^*$ while the trajectories to the left are localized in the chain with $c \approx 1.5$.

How do the trajectories of the driven process at the critical value s^* where the SCGF develops a kink look like? The driven process simulates trajectories corresponding to $c = \Psi'(s^*)$ that is a weighted average of the values in the two phases, and these trajectories are the most likely ones of the unlikely URW trajectories that sample c. In correspondence to equilibrium phase transitions where most often a correlation length scale diverges, I am looking in this case of a dynamical version of a phase transition for a diverging time scale [14]. Whitelam [23] argues that the DPT should be studied by phenomenology, that is finding the mechanism that leads to the singularity in the SCGF which may break the large deviation principle at that point, such that the driven process would not be defined. A candidate I focus on is intermittency and an intrinsic time scale of the trajectories which I call the mean waiting time (MWT) to hop from a phase to another.

Before studying the ER graph model, I look at two simpler models to check for intermittency and the connection of the MWT to the transition region. When I increase the length of the trajectories proportional to the MWT such that the large deviation principle holds again, then this rescales the SCGF around the transition point and the kink in the limit disappears. The first

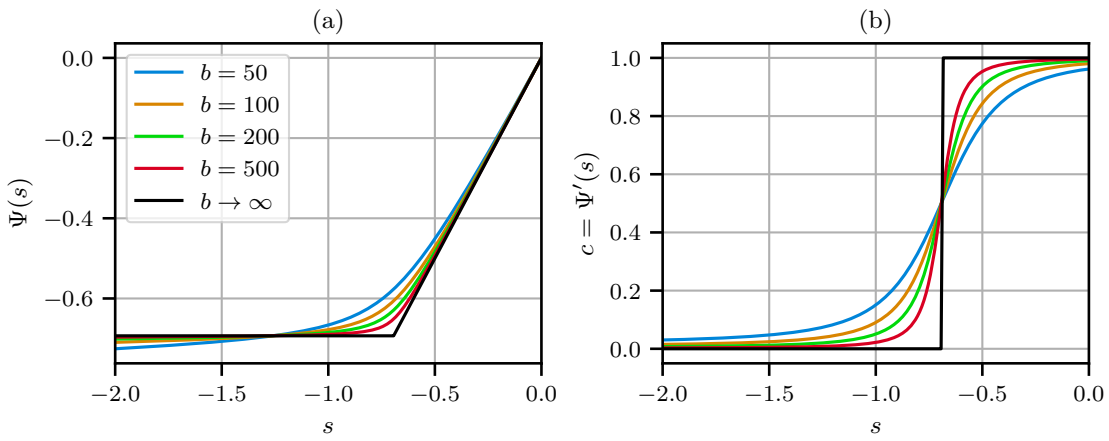


Figure 3.1: The SCGF $\Psi(s)$ and its first derivative for the 2-state model. The system parameter b is increased to get a sharper transition of c around $s^* = -\log(2)$ and thus a kink in Ψ.

model is a reduction of the graph into two states representing the chain and the bulk. After that I study a 4-state model which is based on a fully connected graph that has one dangling chain attached. In the last section I will come back to the ER graph and apply the insights gained from the simpler models.

3.1 2-State Model

The most coarse grained model for the two phases seen in the ER graph is a **2-state model** introduced by Carugno et al. [24]. The state 1 corresponds to the chain and contributes an observable value of 1 while the other state b is the bulk whose contribution is $b > 1$. The assumptions for the transition probabilities are set such that the probability to go from the chain to the bulk is always $\frac{1}{2}$ while the probability to go from the bulk to chain is $\frac{1}{b}$. So for larger b the probability to move out of the bulk becomes lower. This is reasonable as in the ER graph the bulk grows as N increases while the chain stays roughly the same in size. For a sketch of the 2-state Markov chain see figure 3.2. The two circles represent the two states while the arrows indicate the transitions labeled with their transition probability.

The observable defined in the 2-state model is

$$C_t = \frac{1}{t} \sum_{l=0}^{t-1} f_2(X_l), \tag{3.1}$$

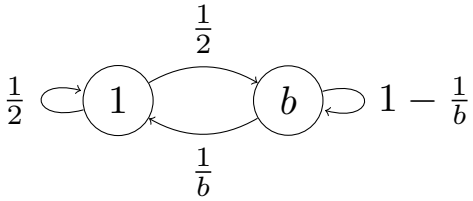


Figure 3.2: Sketch of the 2-state model and its transition probabilities. The system parameter is $b \in (1, \infty)$. The labels on the edges represent the transition probabilities between the states (circles).

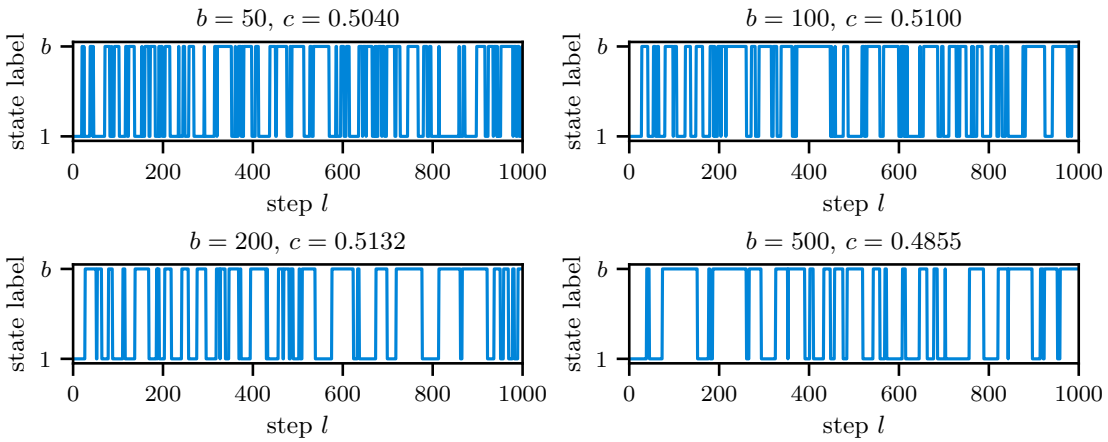


Figure 3.3: Trajectories of the 2-state model simulated with the driven process at $s^* = -\log(2)$ where $c \approx 0.5$. The system parameter b is increased between the subplots.

where

$$f_2(X) = \begin{cases} \frac{1}{b}, & X = 1 \\ 1, & X = b. \end{cases} \tag{3.2}$$

such that $C_t \in [0, 1]$ is a scaled mean degree.

The SCGF Ψ of this model can be with low computational cost computed via direct diagonalization. The value of the observable as a function of the tilting parameter s is obtained by taking the derivative of Ψ. Both functions are shown in figure 3.1 where $s \in [-2, 0]$ and b is increased from 50 to 500. As b gets larger, the SCGF develops a kink at $s^* = -\log(2)$ [24] and the curve of c approaches a step function (black curve). For finite but large b, when there still exist a transition region, s^*, the position of the maximum of Ψ'', already coincides with the analytical value. The value of c at the transition point s^* is $c = 0.5$, which means that the random walk needs to spend equal amounts of time in both states. This is because the rewards of both states are $\frac{1}{b} \approx 0$ and 1 as $b \to \infty$.

The trajectories simulated with the driven process at criticality are shown in figure 3.3. The trajectory length is $t = 10\,000$ of which only the first 1000 steps are shown. In all cases the random walk transitions back and forth between the two states and the fluctuation value c is around 0.5. As b increases one observes that the time between the transitions becomes on average longer. A quantity characterizing the scaling is the **mean waiting time** (MWT) which is the average of the time segments that the random walk spends waiting in one state before it finally moves to the other state. Given a trajectory, the MWT is easily computed by the total time spent in the state divided by the number of transition away from that state.

The MWT over a range of b values from 100 to $20\,000$ is shown in figure 3.4. It is the average over 100 trajectories and the error plotted as the standard deviation (shaded area) is very low and barely visible. Both curves for the MWT of the phase 1 and b follow the same form $\sqrt{b}$ (dashed black curve). The agreement between both phases is expected because the time spent in both states is almost the same. Also the total occupation times of the two states are complementary and the number of transitions from each state can maximally differ by one.

The scaling $\sqrt{b}$ related to the kink in the SCGF was previously derived by Carugno et al. [24]. The new insight is the connection of the MWT to this scaling which you can think of by rescaling

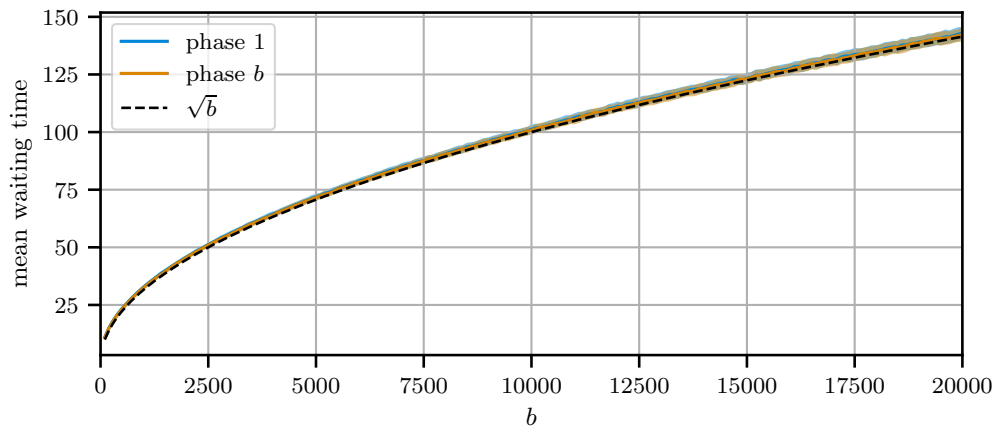


Figure 3.4: Mean waiting time (MWT) for the 2-state model at criticality. The MWT is averaged over 100 trajectories simulated by the driven process at $s^* = -\log(2)$, where the standard deviation is represented by the shaded area. The trajectory length is $t = 10^6$ time steps.

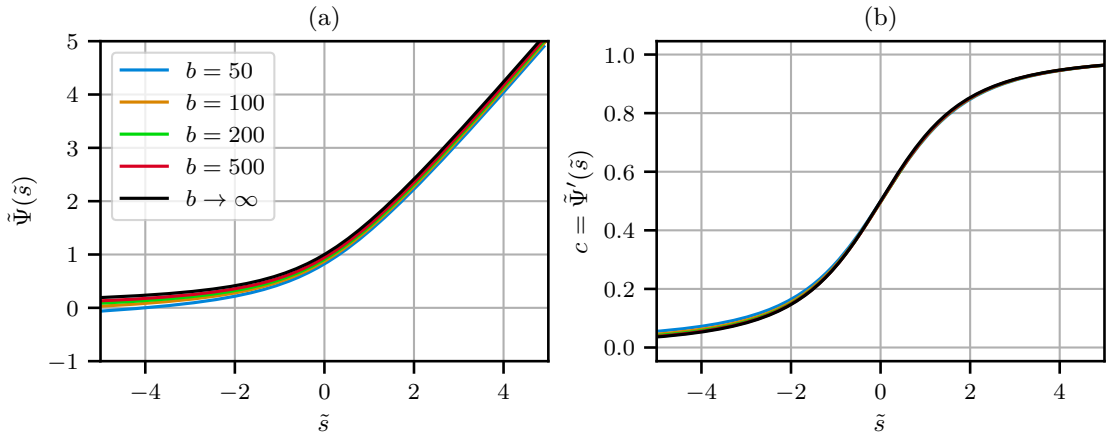


Figure 3.5: Rescaled SCGF $\tilde{\Psi}(\tilde{s}) = \sqrt{b}(\Psi(s(\tilde{s})) - \Psi(s^*))$ and $\tilde{s}(s) = \sqrt{b}(s - s^*)$ for the 2-state model with $\Psi(s^*) = s^* = -\log(2)$.

the time in figure 3.3 so that the MWT becomes constant. Figure 3.5 shows the SCGF which is rescaled around s^* by the factor $\sqrt{b}$. The rescaling to new coordinates

$$\tilde{s}(s) = \sqrt{b}(s - s^*) \tag{3.3}$$

$$\tilde{\Psi}(\tilde{s}) = \sqrt{b}(\Psi(s(\tilde{s})) - \Psi(s^*)), \tag{3.4}$$

where $\Psi(s^*) = s^* = -\log(2)$, maps the SCGF to a new curve whose shape does not drastically change with b and instead converges to a smooth function as $b \to \infty$ (black). In the right plot (b) of figure 3.5 you can see that also the derivative of the rescaled SCGF stays continuous for all b.

3.2 4-State Model

The next step towards the ER graph is a **4-state model** or bulk-dangling-chain model. Here, the bulk is a fully connected graph of $N - 2$ nodes which has a dangling chain of length 2 glued to it [24]. The total number of nodes of that artificial graph is N. Since the bulk is fully connected all nodes in it are equivalent and indistinguishable (degree $N - 3$) except the one which is connected

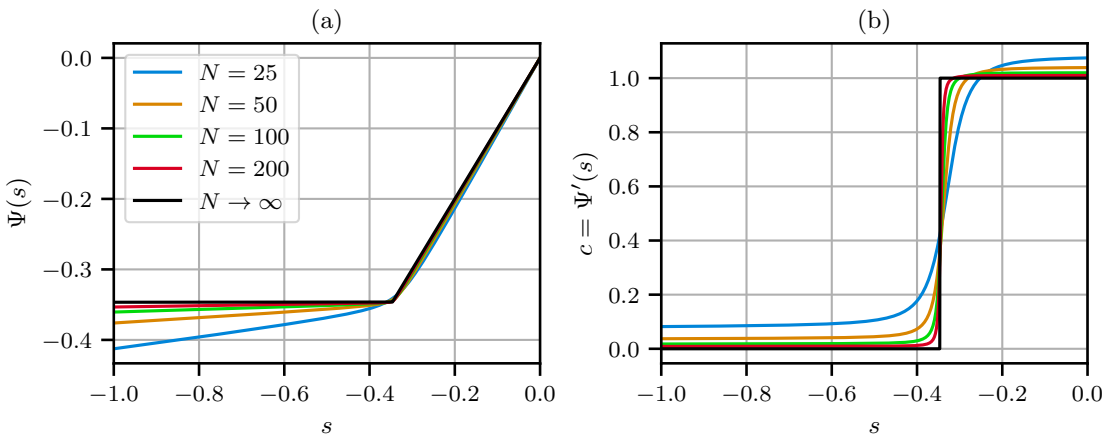


Figure 3.6: The SCGF and its first derivative for the 4-state model. The system parameter N is increased to get a sharper transition of c around $s^* = -\frac{\log(2)}{2}$.

to the chain (degree $N-2$). This means the bulk can be reduced into two states: the group of $N-3$ nodes of degree $N-3$, which has a non-zero probability in transitioning to itself, and the one node of degree $N-2$ which has the connection to the dangling chain. This allows to treat the model by considering only 4 states grouped by their degrees: two for the chain (degree 1 and 2) and two for the bulk (degree $N-2$ and $N-3$). See figure 3.7 for a sketch of the 4-state Markov chain. Nodes are labeled by their degree and the different sizes of the nodes make it easier to distinguish between the chain (small) and the bulk (large).

Similar to the previous section the observable defined in the 4-state model is

$$C_t = \frac{1}{t}\sum_{l=0}^{t-1} f_4(X_l), \tag{3.5}$$

where

$$f_4(X) = \begin{cases} \bar{N}^{-1}, & X = 1 \\ 2\bar{N}^{-1}, & X = 2 \\ (N-2)\bar{N}^{-1}, & X = N-2 \\ (N-3)\bar{N}^{-1}, & X = N-3 \end{cases} \tag{3.6}$$

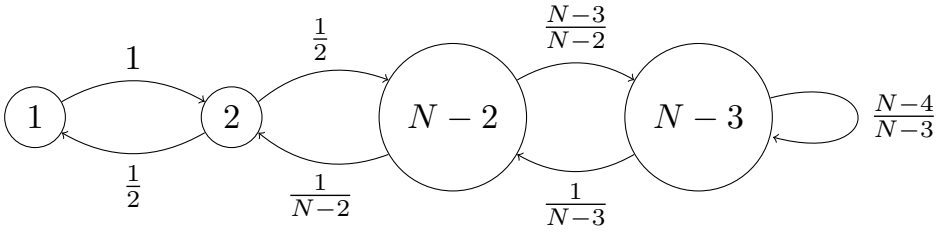


Figure 3.7: Schematics of the 4-state model and its transition probabilities. The system parameter is N, i. e. the number of nodes in the bulk-dangling-chain model. Nodes are labeled by their degree in the bulk-dangling-chain model and the different sizes of the nodes are used to distinguish between the chain (small) and the bulk (large).

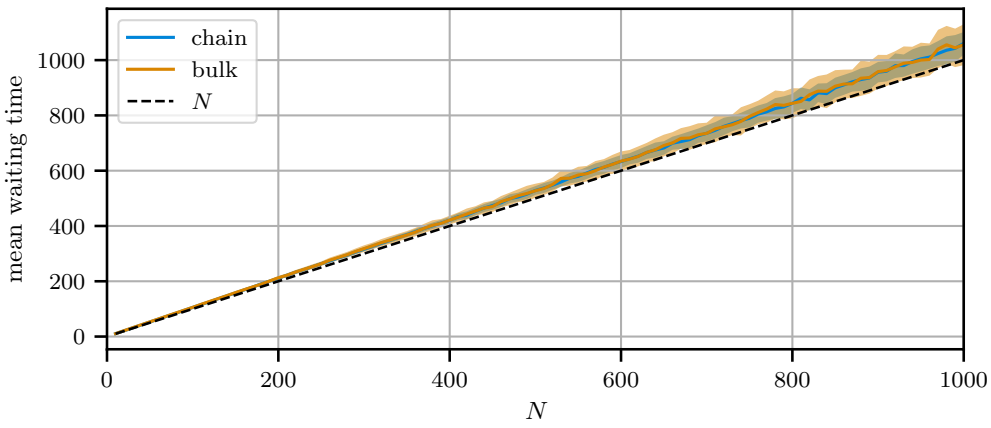


Figure 3.8: Mean waiting time (MWT) of the 4-state model at criticality. The MWT is averaged over 100 trajectories simulated by the driven process at $s^* = -\frac{\log(2)}{2}$, where the shaded are represents the standard deviation. The trajectory length is $t = 10^6$ time steps.

is a scaled degree because $\bar{N} = \frac{1}{N}((N-3)^2 + (N-2) + 2 + 1)$ is the averaged degree of the graph. As $\bar{N}$ scales proportionally to N for large N, it follows that $C_t \approx 1$ in the bulk and $C_t \approx 0$ in the chain.

The SCGF of the 4-state model is computed via exact diagonalization of the titled matrix and plotted in figure 3.6 (a). The interesting range is $s \in [-1, 0]$ as the critical parameter value is $s^* = -\frac{\log(2)}{2}$ [24]. The graph size N shown in figure 3.6 is doubled in steps from 50 to 200. Already for $N = 100$ one sees a kink appearing in the SCGF and the derivative of Ψ, plot (b), becomes almost a step function.

Analogous to the previous section 3.1, I look at the behavior of the MWT in the bulk and the chain. The driven process is simulated with $s^* = \text{argmax}\{\Psi''(s) : s \in [-1, 0]\}$ over a discrete grid of s. The total time of the trajectories is $t = 10^6$ time steps from which the MWT of the combined chain and bulk is computed. This is repeated 100 times to get an averaged MWT and the standard deviation as the error, and the results are shown in figure 3.8. The range of N is from 10 to 1000. The standard deviation is represented by the shaded area around the curves. Both, the MWT of the chain (blue) and the bulk (orange) scale linearly with N and lie almost perfectly on the dashed black line which shows the scaling with N and slope 1. This agrees with the observation in the 2-state model where the MWT follows the scaling function very closely as well.

The reason why I choose to use s^* as the maximum of the second derivative of the SCGF and not the analytical value can be understood by looking at figure 3.9 (a), where the plot shows the s^* vs. the graph size N. The black dashed line represent the analytical value $s^* = -\frac{\log(2)}{2}$. The curve of s^* is converging to the analytical value from above, but since the graph size is still too small I cannot use the analytical value. The driven process at the transiton point is too sensitive to small changes in s especially as N gets larger. On the other hand, in the 2-state model seen before the difference to the analytical value is much smaller, as I also compute the MWT to larger values of b than of N, so there I use the analytical value of s^* directly.

An indication if the random walk is close enough at the transition is to look a the time spent in the bulk and the chain. The right plot (b) in figure 3.9 shows the empirical occupation measure, see equation (1.8), on the four states of the model. The bulk and the chain are visited by the random walk in equal amounts, which means the empirical occupation measure is 0.5. The distribution on individual nodes is different though. While the chain has an empirical

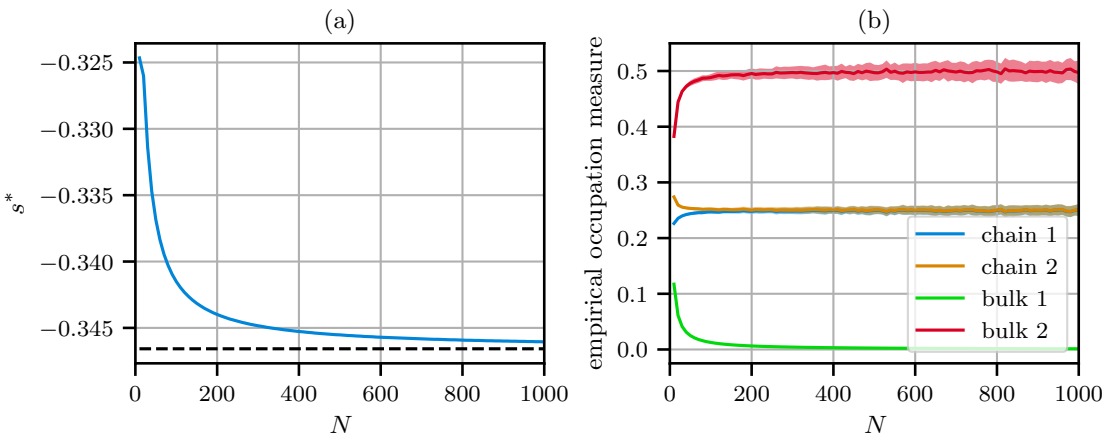


Figure 3.9: Critical parameter s^* and the empirical occupation measure for the 4-state model over different graph sizes N. The length of trajectories is $t = 10^6$ and the empirical occupation measure is averaged over 100 trajectories. The standard deviation is shown as the shaded area around the curves.

occupation measure of about 0.25 on both nodes, the bulk is dominated by the big bulk, labeled *bulk 2*, and the relative time spent in the node connecting the bulk and the chain, labeled *bulk 1*, goes to zero. That the empirical occupation measure is distributed like this over the four states indicates intermittency. Also the trajectories at criticality shown in figure 3.10 indicate intermittent behavior.

Intermittency means that the random walk spends some time in the chain before moving to the bulk where it spends some time before moving back again to the chain. The opposite of intermittency which the random walk could utilize to sample the fluctuation $c = 0.5$ would be to jump back and forth only between the two inner nodes of the chain and the bulk. In that case the occupation measure would be 0.5 on *chain 2* and *bulk 1* while on the other nodes it would be close to zero.

The trajectories in figure 3.10 also show how the rescaling of time corrects the increasing MWT. One can see that the number of transitions between the bulk and the chain stay approximately the same as the time plotted is increased linearly with N. This means that the large deviation principle is restored as the time t is increased simultaneously with N and thus the trajectories look unchanged.

The scaling factor N which is seen in the MWT of the 4-state model (figure 3.8) was also predicted by Carugno et al. [24]. Finally, this diverging time scale is used to rescale the SCGF

$$\tilde{s}(s) = N(s - s^*) \tag{3.7}$$

$$\tilde{\Psi}(\tilde{s}) = N(\Psi(s(\tilde{s})) - \Psi(s^*)) \tag{3.8}$$

to remove the kink. It is $\Psi(s^*) = s^* = -\frac{\log(2)}{2}$ because the SCGF in the limit $N \to \infty$ has the slope of 1 to the right of s^* and goes through the origin. The rescaled SCGF is shown in figure 3.11 (a) and its derivative in (b). For $N \to \infty$ both curves converge to continuous functions meaning that the large deviation principle holds again with the new rescaled time $\tilde{t} = tN$.

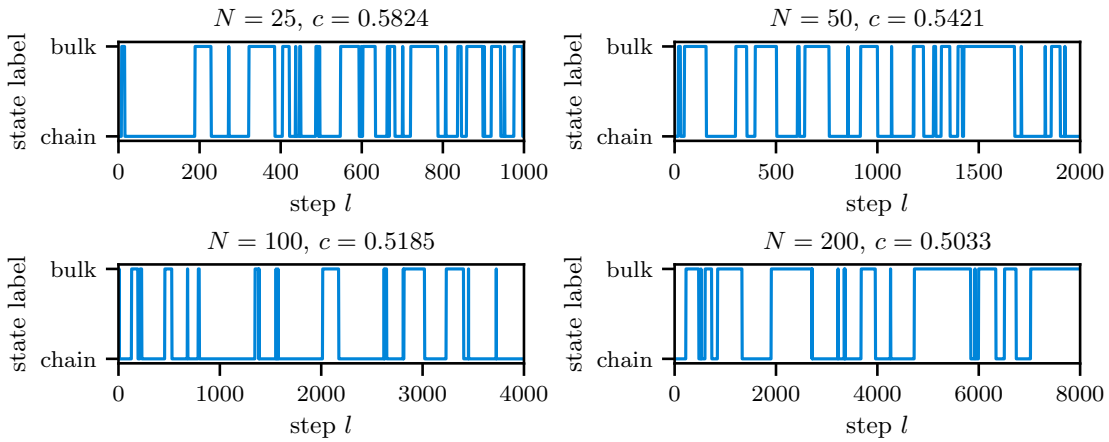


Figure 3.10: Trajectories of the 4-state model for different values of N simulated by the driven process with $s^* = \text{argmax}\{\Psi''(s) : s \in [-1, 0]\}$. The plotted time is increased linear with N.

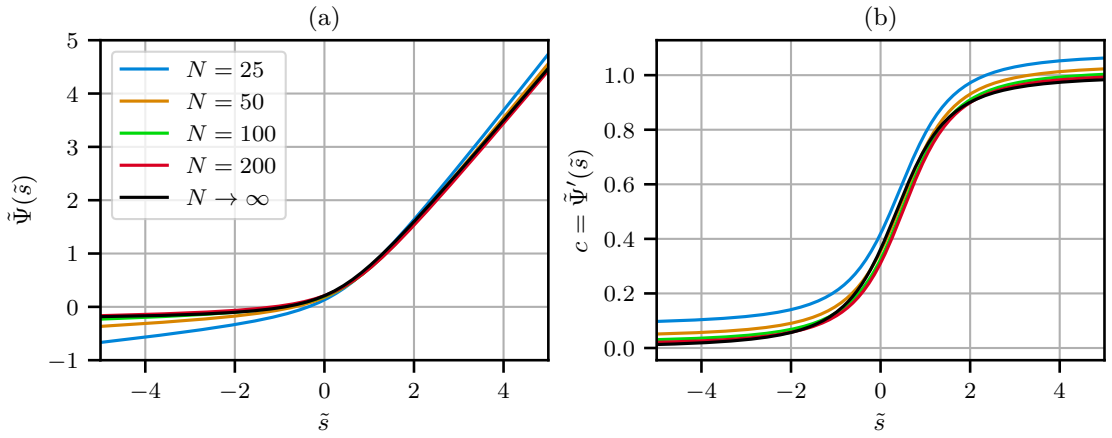


Figure 3.11: Rescaled SCGF $\tilde{\Psi}(\tilde{s}) = N(\Psi(s(\tilde{s})) - \Psi(s^*))$ and $\tilde{s}(s) = N(s - s^*)$ for the 4-state model with $\Psi(s^*) = s^* = -\frac{\log(2)}{2}$.

3.2.1 Intermittency Check

A heuristic check for intermittency which we see above was proposed by Whitelam and Jacobson [14]. The idea is to look at the negative log-probability of different ways to realize a fluctuations c. The function $U(c)$ describes the negative log-probability of a two-step driven process at criticality, i. e. with the transition matrix Π_{s^*}. I look at three cases with different fluctuations: starting and staying in the chain $c = 0$, going from the chain to the bulk to the chain and vise versa $c = \frac{1}{2}$, and starting and staying in the bulk $c = 1$. In terms of the transition matrix elements $\Pi_{s^*}(i, j)$ the $U(c)$ are

$$U\left(\frac{1}{2}\right) = -\frac{1}{2}\log\left(2\Pi_{s^*}(2,3)\Pi_{s^*}(3,2)\right) \tag{3.9}$$

$$U(0) = -\frac{1}{2}\log\left(2\Pi_{s^*}(1,2)\Pi_{s^*}(2,1)\right) \tag{3.10}$$

$$\begin{aligned} U(1) = -\frac{1}{2}\log\Big(&\Pi_{s^*}(4,4)\Pi_{s^*}(4,4) + 2\Pi_{s^*}(4,3)\Pi_{s^*}(3,4) \\ &+ \Pi_{s^*}(4,4)\Pi_{s^*}(4,3) + \Pi_{s^*}(3,4)\Pi_{s^*}(4,4)\Big). \end{aligned} \tag{3.11}$$

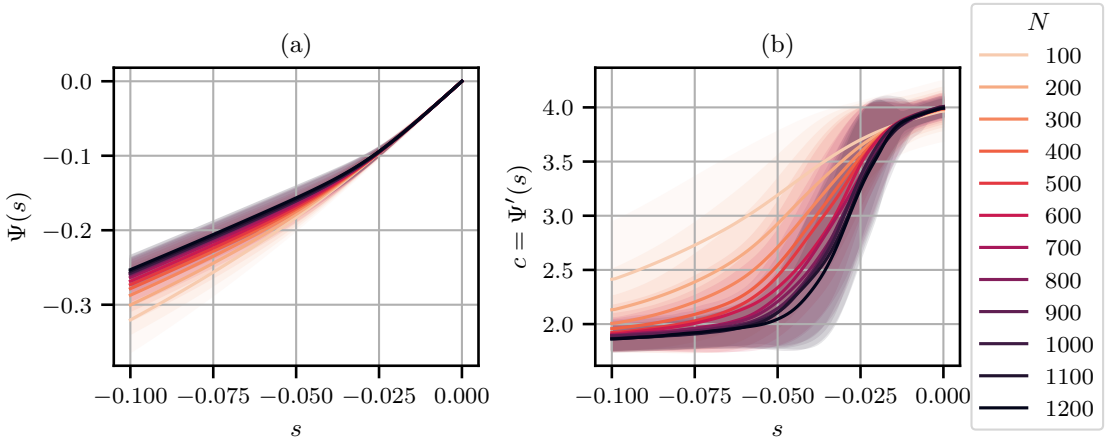


Figure 3.12: Averaged SCGF of the ER graph for graph sizes N ranging from 100 to 1200. The shaded area shows the standard deviation over the averaging of 1000 graph realizations for $N \leq 500$ and over 200 for $N \geq 600$.

The labelling of the states is 1 to 4 corresponding from left to right in figure 3.7. Intermittency is expected if $\frac{U(0)+U(1)}{2} < U(\frac{1}{2})$. This means that spending all the time on the two inner nodes and sampling $c = \frac{1}{2}$ is less likely than spending some time in the chain and the bulk and switching in between. This is the case for the 4-state model as

$$U(0) \approx 2.65 \cdot 10^{-3} \tag{3.12}$$

$$U(1) \approx 2.65 \cdot 10^{-3} \tag{3.13}$$

$$U\left(\frac{1}{2}\right) \approx 2.82 \cdot 10^{0}. \tag{3.14}$$

3.3 Erdős-Rényi Graph

In this last section I come back to the ER graph, where figure 3.12 shows the averaged SCGF and its first derivative. The averaging is done for a fixed N over many graph realizations (1000 for $N \leq 500$ and 200 for $N \geq 600$). The shaded area represents the standard deviation of the averaging.

One sees that the SCGF curves lie closer together for larger N. They develop a transition point (believed to become a kink for $N \to \infty$) which is seen more clearly by looking at the first derivative $c = \Psi'(s)$ which transitions from around 2 to 4 in the range $s \in [-0.1, 0]$. The transition in c becomes sharper with increasing N although slower than it is seen in the two previous models. The difficulty with the ER graph is that the individual graph realizations are created randomly and the position of the transition point in the SCGF is not fixed. Thus the averaging smoothens out the transition especially for smaller N.

The critical value s^* where c has the steepest increase can be computed individually for all the SCGFs contained in the averaging. The average of these s^* is shown in figure 3.13 with the error bars representing the standard deviation. In comparison the orange cross markers represent the s^* obtained from the averaged SCGF seen in figure 3.12 (a). Both estimates of s^* as a function of N seem to flatten out for $N > 700$ although the uncertainty is still significant. Also the s^* from the averaged SCGF lies above the average of the individual s^* values. Nevertheless, the averaging helps in restricting the possible region where the critical s^* lies in. With this knowledge

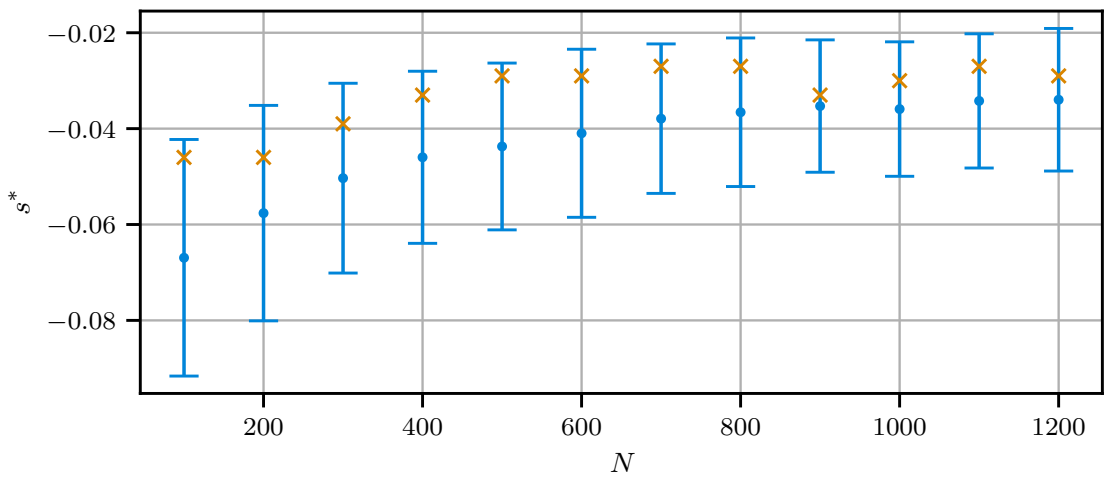


Figure 3.13: Critical value s^* vs. the graph size N. The black dots are the averages of the individual s^* computed over 100 graph realizations and error bars are the corresponding standard deviations. The orange crosses are the s^* computed from the averaged SCGF.

it becomes easier to simulate the driven process as I fix s to a certain value in that range and select the graphs for which the value is close enough to the transition point, i. e. $s^* \approx s$.

The criterion for estimating that the driven process is at the transition is to look at the stationary distribution coarse grained into the chain and bulk. This is done before doing the simulation as the construction of the driven process returns the stationary distribution over the nodes, see section 1.4. After the simulation of the trajectory (length $t = 10^7$) the criticality is checked again by computing the coarse grained empirical occupation measure.

For the coarse graining I first need to group the nodes of the ER graph into bulk and chain. For the chain I consider all dangling chains of the ER graph. These are all the nodes of degree $k \leq 2$ that are in the chains. The rest of the nodes of the graph make up the bulk. Special consideration is needed to treat the nodes connecting the bulk and the chains which I call the gateways. Here I applied a variable scheme where a gateway is accounted to the bulk if it is visited from the bulk and as the chain if it is visited from the chain. Thus the random walk only changes the phase if it transitons through the gateway from the bulk to the chain or vice versa.

In figure 3.14 segments of three trajectories are shown at different values of s, but for fixed graph size $N = 400$. From left to right the plots are first above criticality $s > s^*$, at criticality $s \approx s^*$ and below criticality $s < s^*$. The trajectories above and below criticality show that the random walk spends most of the time in, respectively the bulk and the chain. Once in that phase it only occasionally visits the other phase. The values of the observable c in these cases are dominated either by the bulk or the chain. The trajectory at criticality, see middle plot for $c = 2.64$, is different. Here the random walk transitions quite frequently back and forth between the bulk and the chain so it samples an intermediate value of c. Below I list the computed empirical occupation measure on the bulk and the chain for the trajectories together with the values of c and s.

bulk	chain	c	s
0.895	0.105	3.54	−0.020
0.504	0.496	2.64	−0.042
0.004	0.996	1.70	−0.200

In the 4-state model (section 3.2) the driven process is at the transition point when the empirical

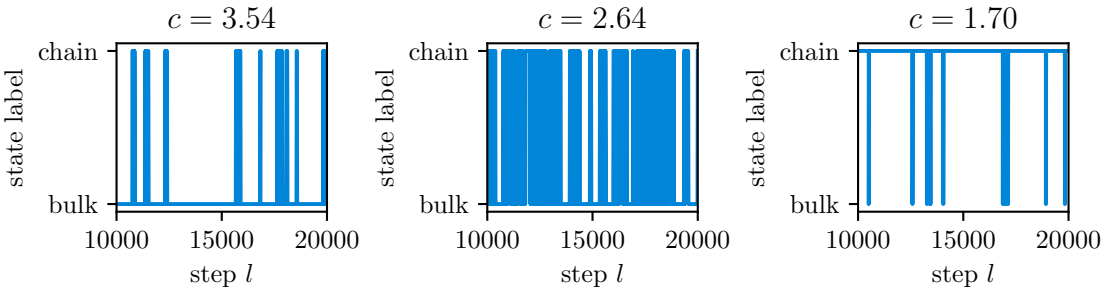


Figure 3.14: Trajectories of the driven process for a graph size of $N = 400$. The values of s are from left to right -0.020, -0.042 and -0.2 sampling different values c. The total length of the trajectories is $t = 10^7$ steps of which here is shown a segment of 10 000 steps.

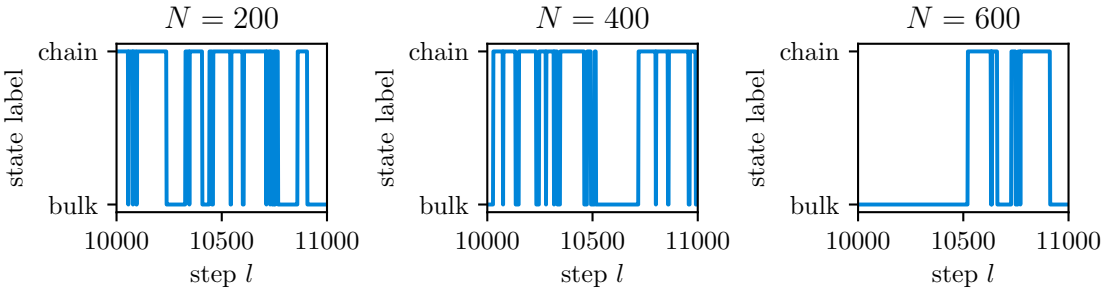


Figure 3.15: Trajectories at criticality of the driven process for three different graph sizes N. The total length of the trajectories is $t = 10^7$ of which a segment of length 1000 is shown. The value of s^* is from left to right -0.054, -0.052 and -0.050 with corresponding c values 2.55, 2.55 and 2.67.

occupation measure in the bulk and the chain are equal. It becomes clear that also for the ER model the driven process is at the transition point when the occupation measure is balanced, i. e. 0.5 for bulk and chain. This criterion is much more reliable than looking at the value of c at s^* as this depends on the minimum and maximum degree of the realized graph.

Similar to the critical trajectories of the 4-state model in section 3.2 the trajectories on the ER graph show intermittency meaning that there are quite long stretches where the random walk stays in one phase. Figure 3.15 shows trajectories ($t = 10^7$, but only a segment of 1000 steps is plotted) of the driven process for three different graph sizes $N = 200$, 400 and 600. The value of s^* for these simulations is computed by finding the peak in the first derivative of the stationary distribution coarse grained on the chain. This method is numerically more stable and practical than looking at the second derivative of the SCGF as the latter one can have multiple maxima. The achieved empirical occupation measures in the bulk and the chain are as desired very close to 0.5, see below.

N	bulk	chain	c	s^*	MWT bulk	MWT chain
200	0.493	0.507	2.55	-0.054	49	51
400	0.504	0.496	2.55	-0.052	67	66
600	0.507	0.493	2.67	-0.050	63	61

The mean waiting times (MWT) are computed as the total number of time steps spent in each region (bulk or chain) divided by the number of transitions away from each region. Since there are only two regions the number of transitions are almost always equal and can only differ by one. Thus, as the empirical occupation measures on both phases are very close also the MWTs are almost equal.

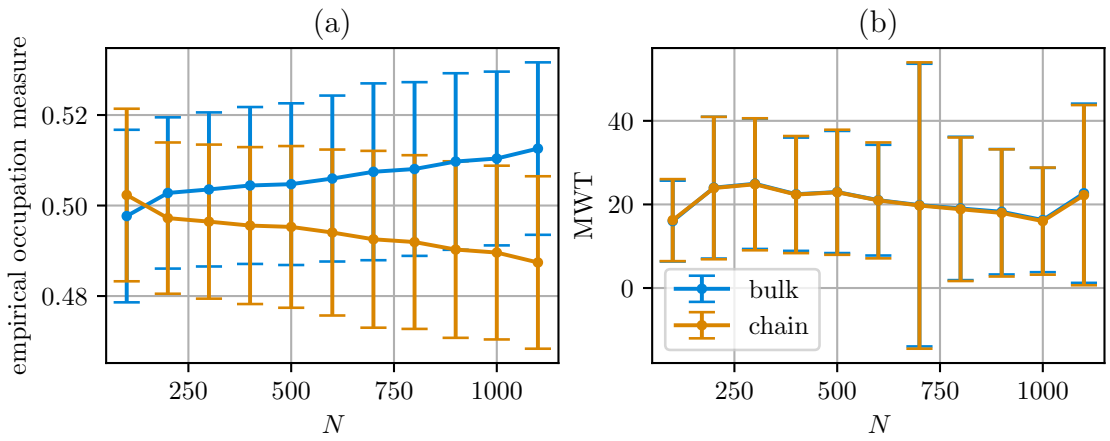


Figure 3.16: The empirical occupation measure and the MWT for the ER graph model. Averages are over 900 to 1000 graph realizations where the empirical occupation measure in the bulk and chain is between 0.45 and 0.55. The error bars represent the standard deviation.

Based on the observation that the MWT computed for an individual realization does not show a dependence on N I apply the stationary distribution check to generate many graphs and average the obtained MWTs. For this I fix the value of s^* to the averages shown in figure 3.13 and only compute the MWT for graphs where the empirical occupation measure in the bulk and chain is within the interval $[0.45, 0.55]$ as described above. This yields between 900 to 1000 accepted graph realizations where the overhead is that about 10 times more graphs are generated.

The averaged empirical occupation measure and the averaged MWT are shown in figure 3.16. The different graphs sizes range from $N = 100$ to only 1100 because of limited computational resources. One can see that the empirical occupation measure, though close to 0.5 gets more unbalanced as N increases indicating that the method of generating graphs for a fixed s^* becomes less precise as N increases. Nevertheless, the MWT in plot (b) is little affected by the slight change in the empirical occupation measure. Quite different to the 2- and 4-state model, the MWT does not show a dependence on N but stays roughly constant within the errors. Also the magnitude of about 20 is much smaller than what the other models predicted.

A drawback of the analysis on the ER model is that N does not span multiple orders of magnitudes. Thus one cannot conclude how the MWT actually scales with N, only that, if it does, is has to be very slow. That the average distances in the ER graph grow with $\log(N)$ [3] could be a possible reason. Further analysis especially with more sophisticated methods in finding the transition point is required in the future.

Conclusion

In this Master thesis I studied fluctuations and rare events of time-additive observables defined on discrete-time Markov chains on finite state spaces. The central quantity is the mean degree of a random walk on an Erdős-Rényi (ER) random graph. I implemented the Adaptive Power Method (APM), a modified random walk, that converges to the driven process and samples large deviations of the observable of interest. The numerical analysis showed that through transfer learning the convergence time of the APM is reduced and the scaled cumulant generating function (SCGF) and rate function become computable from a single trajectory. Further, I investigated the appearance of a dynamical phase transition (DPT) related to the development of a kink in the SCGF. In two simpler models capturing the bulk-dangling-chain properties of the ER graph it was found that the DPT is caused by intermittency in the trajectories and that the mean waiting time (MWT) in the phases yields the correct rescaling of the SCGF in order to remove the kink.

Future work is necessary to improve the APM, especially the fact that it needs to visit the whole graph when $s < 0$ before sampling the correct fluctuation. Further development on the APM conditioning scheme on a fluctuation could increase the applicability of the APM to real world networks for studying the impacts of rare events. An open question on the DPT that remains is the scaling of the MWT in the ER graph model. Using more advanced eigenvalue solvers could help going to larger graphs sizes, but also simpler models which capture the same scaling could be considered. Once again, we experience that even easily constructed problems, like the ER graph and a random walk, show a rich behavior and give us a vast playground to ask questions. Just as Golan Trevize, we learn much more on the way than at the final destination.

The source code of the adaptive power method is available under
https://github.com/dastu08/adaptive-power-method.

Bibliography

[1] M. Newman, *Networks: an introduction* (Oxford University Press, Oxford, Mar. 2010).

[2] A. Barrat, M. Barthlemy, and A. Vespignani, *Dynamical processes on complex networks* (Cambridge University Press, Cambridge, 2008).

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: structure and dynamics", Phys. Rep. **424**, 175 (2006).

[4] N. Van Kampen, *Stochastic processes in physics and chemistry*, 3rd ed. (Elsevier, Amsterdam, 2007).

[5] H. Touchette, "The large deviation approach to statistical mechanics", Phys. Rep. **478**, 1 (2009).

[6] F. Coghi, J. Morand, and H. Touchette, "Large deviations of random walks on random graphs", Phys. Rev. E **99**, 022137 (2019).

[7] R. L. Jack, "Ergodicity and large deviations in physical systems with stochastic dynamics", Eur. Phys. J. B **93**, 74 (2020).

[8] R. L. Jack and P. Sollich, "Large deviations and ensembles of trajectories in stochastic models", Prog. Theor. Phys. Supp. **184**, 304 (2010).

[9] V. S. Borkar, S. Juneja, and A. A. Kherani, "Performance analysis conditioned on rare events: an adaptive simulation scheme", Commun. Info. Syst. **3**, 259 (2003).

[10] F. Coghi and H. Touchette, "Adaptive power method for estimating large deviations in markov chains", Phys. Rev. E **107**, 034137 (2023).

[11] G. Ferré and H. Touchette, "Adaptive sampling of large deviations", J. Stat. Phys. **172**, 1525 (2018).

[12] C. D. Bacco, A. Guggiola, R. Kühn, and P. Paga, "Rare events statistics of random walks on networks: localisation and other dynamical phase transitions", J. Phys. A: Math. Theor. **49**, 184003 (2016).

[13] G. Di Bona, L. Di Gaetano, V. Latora, and F. Coghi, "Maximal dispersion of adaptive random walks", Phys. Rev. Res. **4**, L042051 (2022).

[14] S. Whitelam and D. Jacobson, "Varied phenomenology of models displaying dynamical large-deviation singularities", Phys. Rev. E **103**, 032152 (2021).

[15] I. Tishby, O. Biham, E. Katzav, and R. Kühn, "Revealing the microstructure of the giant component in random graph ensembles", Phys. Rev. E **97**, 042318 (2018).

[16] G. R. Grimmet and D. R. Stirzaker, *Probability and random processes* (Oxford University Press, Oxford, 2001).

[17] R. S. Ellis, *Entropy, large deviations, and statistical mechanics* (Springer, New York, 1985).

[18] R. Chetrite and H. Touchette, "Nonequilibrium markov processes conditioned on large deviations", Ann. Henri Poincaré **16**, 2005 (2015).

[19] E. Seneta, *Non-negative matrices and markov chains*, 2nd ed. (Springer, New York, 2006).

[20] R. Chetrite and H. Touchette, "Variational and optimal control representations of conditioned and driven processes", J. Stat. Mech.: Theory Exp. **2015**, P12001 (2015).

[21] G. Carugno, P. Vivo, and F. Coghi, "Graph-combinatorial approach for large deviations of markov chains", J. Phys. A: Math. Theor. **55**, 295001 (2022).

[22] V. S. Borkar, *Stochastic approximation* (Hindustan Book Agency Gurgaon, New Delhi, 2008).

[23] S. Whitelam, "Comment on the literature definition(s) of 'dynamical phase transition'", (2021), arXiv:2112.09107.

[24] G. Carugno, P. Vivo, and F. Coghi, "Delocalization-localization dynamical phase transition of random walks on graphs", Phys. Rev. E **107**, 024126 (2023).