# Performance of Averaging Algorithms in Time-Varying Networks

Patrick Denantes

Master's Thesis

Ecole Polytechnique Fédérale de Lausanne
I&C School of Computer and Communication Sciences
LCAV, Prof. Martin Vetterli LCA, Prof. Patrick Thiran

Munich University of Technology
Associate Institute for Signal Processing
Univ.-Prof. Dr.-Ing. Wolfgang Utschick

Date of Start: November 1, 2006
Date of Examination: June 4, 2007

Supervisors:

EPFL: Florence Bénézit
Prof. Patrick Thiran (LCA)
Prof. Martin Vetterli (LCAV)

TUM: David Schmidt
Prof. Wolfgang Utschick

EPFL, CH-1015 Lausanne May 30, 2007

**Acknowledgments**

I would first like to thank Prof. Rüdiger Urbanke from the LTHC and all the people from the Information Processing Group (IPG) at EPFL who kindly hosted me in their Lab during my stay in Lausanne. Further thanks go to Florence, for all the interesting and insightful discussions we had. It was a great pleasure to work with her in a team. Many thanks go to Prof. Thiran who guided and supervised our work, and reviewed a previous version of this document. I would also like to thank Prof. Vetterli who followed our research efforts through numerous private meetings. Last but not least, special thanks go to Prof. Utschick for giving me the opportunity to conduct this work at EPFL, and also to David for the discussions we had during my visits back at TU Munich, and who helped correcting and improving the readability of this paper.

# Abstract

We study averaging algorithms in time-varying networks, and means to measure their performance. We present sufficient conditions on these algorithms, which ensure they lead to computation at each node, of the global average of measurements provided by each node in the network.

Further, we present and use results from ergodic theory to define an accurate performance metric for averaging algorithms. This metric, the *contraction coefficient*, differs from previously used metrics such as the second largest eigenvalue of the expected weighting matrix, which gives an approximation of the real convergence rate only in some special cases which are hard to specify.

On the other hand, the contraction coefficient as set forth herein characterizes exactly the actual asymptotic convergence rate of the system. Additionally, it may be bounded by a very concise formula, and simulations show that this bound is, at least in all studied cases, reasonably tight so as to be used as an approximation to the actual contraction coefficient.

Finally, we provide a few results and observations which make use of the derived tools. These observations may be used to find new optima for design parameters of some averaging algorithms, and also open the door to new problems in the study of the underlying mathematical models.

# Contents

# Introduction

The problem of distributed coordination in networks has applications in all systems in which a number of "agents" need to compare or to align an internal "measure" with the measures of the other agents. A common example of a set of agents is a network of autonomous sensors, which need to compare their own measurement (e.g. temperature) to the measurements of other sensors, in order to detect an anomaly (e.g. a fire). Sensor networks may be used in a variety of applications, such as environment monitoring, homeland security, or traffic control, to give only a few examples [1]. Distributed coordination problems also arise in congestion control of data networks, load balancing in distributed computing architectures, clock synchronization, in formation control of mobile units (also known as flock control) [2, 3], or attitude alignment of satellite clusters. In all these distributed systems, there is a need for the individual agents to know the average of some measure (e.g. traffic flow, CPU load, time, position, heading, attitude) over all other agents (e.g. routers, processors, clocks, robots, autonomous vehicles, satellites), so they can adjust their behavior and/or coordinate their effort. Further, the averaging problem is a first step towards more sophisticated distributed algorithms. For example when measurements are linear in the variable of interest, and with additive Gaussian noise, averaging protocols may be used to implement a distributed maximum likelihood estimator [4].

In many cases, the network may be changing over time, for various reasons. Nodes may join or leave the network, or become inactive or unavailable to resume operation later. Further, also the connections between agents may fail for various reasons, such as channel fading, upcoming obstacles or broken transmission systems. Thus, robust averaging algorithms are needed, which can cope with a varying network topology, and even with a varying number of nodes. There are known algorithms which satisfy these conditions, such as *gossip algorithms* [5, 6, 7] and *synchronous averaging algorithms* [8, 9]. However, there is a lack of tools to analyze the performance of these algorithms, in order to properly compare their performance. Further, their mechanism is often not fully understood, which prevents a systematic optimization of design parameters. The time-variability and randomness involved in the averaging process due to failing links, nodes, changes in network topology or size, make it difficult to analyze performance of these algorithms, yet even to specify a suitable performance metric.

Therefore, the first goal of this work is to provide means to easily compare, and understand the operation principles of these algorithms. This is achieved by the specification of an *contraction coefficient*. It is shown how this value can be computed by simple means. Further, bounds guaranteeing a minimal contraction rate are presented. Finally, experimental results illustrate the relevance of this metric, and show how it can be used to optimize design parameters of averaging algorithms.

**Structure of the Document**

We begin by presenting different types of known robust averaging techniques, including gossip algorithms and different types of synchronous averaging. These are all iterative algorithms, designed such that an estimate of the value of interest (i.e. average) sequentially updated at each node tends to the actual value. A common notation which allows to describe all the different algorithms con-

sidered in the same way and to study their performance, or convergence speed, is described in Section 1.3 and used in the sequel.

Chapter 2 addresses the question whether an averaging algorithm as described in the first chapter can produce an accurate solution at all. This problem is viewed from the point of view of renewal theory. Sufficient conditions for sequences of average estimates computed at each node to tend to the desired value are derived in Section 2.1. Also, it is shown that a measure for the estimation error decays exponentially with algorithm iterations.

To characterize and understand this exponential contraction of the error requires some mathematical background. These notions are presented in Chapter 3, which is kept in a much more general setting. It presents important theorems from ergodic theory and their application to products of random matrices, a special case of which describes our averaging processes. In particular, we introduce the notion of *contraction coefficient*, which we will use to measure the performance of averaging algorithms.

In Chapter 4, in an attempt to compute or to approximate this contraction coefficient in a simple way, we derive two upper bounds on it. One of these turns out to be fairly tight, while the other requires less effort to compute, but is always looser than the first, and in most cases significantly looser.

This is illustrated by simulations which results are set forth in Chapter 5. We studied the performance of different averaging schemes in different network topologies, as well as the tightness and usefulness of the bounds derived in the previous chapter.

Finally, Chapter 6 points out observations on the behavior and performance of averaging algorithms, which became apparent while using of the newly introduced tools. These observations may lead to a better understanding of the algorithms, and help to adapt them to specific applications, in order to maximize their performance. A concluding section summarizes the results.

# 1. The Averaging Problem

## 1.1 Introduction

We consider a time varying network, whose goal is to make available to each node the average value of the measurements of all nodes in the network, or at least an approximation of it. To this end, at time instant $t$, the nodes can communicate with each other over all currently active graph edges, or communication links. We consider time-slotted algorithms.

We will restrict the messages exchanged to contain only a current, partly averaged, value $x_i(t)$ of the sending node $i$, and in some cases limited information about the degree of the sending node and its neighbors. This also means that the same message is broadcast to all neighbors, regardless on the message received from the receiving node at previous time instants, or even whether the receiving node was previously a neighbor at all. Algorithms in which different, customized, messages are sent to each neighbor, such as described in [10, 11], are known as *Consensus Propagation* algorithms, and are not dealt with in this document.

Further, the processing of the messages received at each node is restricted to linear operations. More specifically, at each time step, every node $i$ performs an update operation on its current estimate $x_i(t)$ of the overall average. This operation is linear, and relies only on the current average estimates from node $i$ and its neighbors. The update equation for node $i$ at time $t$ then reads as

$$x_i(t+1) = w_{ii}(t)x_i(t) + \sum_{j\in\mathcal{N}_i} w_{ij}(t)x_j(t) \qquad i = 1\dots n \tag{1.1.1}$$

where $w_{ij}(t)$ denote weighting factors which values are set according to some algorithm to be defined, and $\mathcal{N}_i$ is the neighborhood of node $i$, i.e. the set of nodes which have link to node $i$. A link between nodes $i$ and $j$ may be down, or broken at time $t$, in which case $w_{ij}(t) = 0$. $x_i(0)$ is the initial measurement at node $i$.

In a network of $n$ nodes, our goal is to compute the average

$$\bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i(0). \tag{1.1.2}$$

Before specifying the exact values of the weighting factors, it is useful to set up the following two conditions on these numbers:

$$\sum_j w_{ij}(t) = 1 \qquad \forall t \geq 0,\ \forall i = 1\dots n \tag{1.1.3}$$

$$\sum_i w_{ij}(t) = 1 \qquad \forall t \geq 0,\ \forall j = 1\dots n. \tag{1.1.4}$$

Condition (1.1.3) ensures that the fully averaged configuration $x_i = y\ \forall i$ is a fixed point of the update equation (1.1.1), and condition (1.1.4) ensures the conservation of sum. That is, at any given time $t$, $\sum_i x_i(t+1) = \sum_i x_i(t)$, and if we reach a fixed point of the form described above, we will actually have $y = \bar{x}$.

These conditions are also found e.g. in [12], and can be fulfilled by many different specific assignments for the weights $w_{ij}$, a couple of which are described below.

## 1.2 Averaging Algorithms

The following list is not supposed to be exhaustive, and is rather meant to give a couple of examples how the weighting factors may be chosen. It is to be noted that in all the algorithms described below lead to symmetric weighting factors, that is $w_{ij} = w_{ji}\ \forall i, j$. This is not necessary, but very convenient for fulfilling conditions (1.1.3) and (1.1.4), as both are then equivalent.

### 1.2.1 Uniform Weights

Also called the *maximum-degree* method, this is the simplest way of choosing the weighting factors. Take the weights all equal to each other on the edges, and fill up with the self-loop weight so they add up to one. That is,

$$w_{ij}(t) = \begin{cases} \alpha & \text{if } j \in \mathcal{N}_i(t) \\ 1 - \sum_{k \in \mathcal{N}_i(t)} w_{ik}(t) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{1.2.1}$$

where $\alpha$ is a constant. If it is less than one over the maximum degree $d_{\text{max}}$ of all nodes at all times, the weights on the self-loops $w_{ii}(t)$ are always positive, and the algorithm is stable, i.e. an error measure[1] on the $x_i(t)$ is non-increasing. For that reason, the most commonly used choice is $\alpha = 1/(d_{\text{max}} + 1)$.

When $\alpha = 1/d_{\text{max}}$, the weights on the self-loops $w_{ii}$ can be zero. This can lead to a problem in the following situation. Consider the case of a regular bipartite graph, in which all links are always active, so the weights do not depend on time. As the graph is regular, $d_i = d_{\text{max}}$ and $w_{ii} = 0$ for all $i$. Because the graph is bipartite, at each time step, all nodes on one side will compute a weighted sum of values from nodes on the other side only, and vice-versa. In this way, the values from one side will never get averaged with the values on the other side, and the algorithm will therefore never reach the desired fixed point.

For $\alpha > 1/d_{\text{max}}$, some self-loops $w_{ii}$ may be negative, and the system may become unstable, causing some $x_i(t)$ to diverge.

The most conservative choice would be of course $\alpha = 1/n$, which might be used if the maximum degree is unknown. However, this leads to very small weights if the network is large and is therefore not very effective. Moreover, this assumes knowledge of the total number $n$ of nodes in the network.

Note that knowledge of $d_{\text{max}}$ at every node is in general not a trivial assumption either. But in technical systems, it is easy to limit the number of links to a node to a number given a priori.

### 1.2.2 Metropolis Weights

The Metropolis weights method is inspired from the Metropolis–Hastings algorithm used in Markov chain Monte Carlo methods [13]. On a time-varying graph, the weights are defined as

$$w_{ij}(t) = \begin{cases} \frac{1}{1+\max\{d_i(t), d_j(t)\}} & \text{if } j \in \mathcal{N}_i(t) \\ 1 - \sum_{k \in \mathcal{N}_i(t)} w_{ik}(t) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{1.2.2}$$

[1] For example $\max_i |x_i(t) - \bar{x}|$.

So the weight on each edge is one over the larger degree at its two incident nodes plus one, and the self-weights are chosen so as all weights sum up to one at each node. Again, as in the maximum-degree method, one could think of dropping the "plus one" term, thus allowing self-weights to be zero and taking the risk not to converge, if the graph happens to be always bipartite.

Note that the Metropolis weights can be computed locally, as this operation requires only knowledge of the own degree and those of direct neighbors. It is realistic to perform it at each time step, unlike the optimization described in [9] and discussed in the previous paragraph. However, every node still needs to transmit its own current degree at each step, which creates a non negligible overhead.

A comparison between maximum-degree and Metropolis weights in [4] shows that the latter converges significantly faster than the former.

### 1.2.3 Optimized Weights

While being easy to use and to implement, the heuristic rules presented above will in general give a suboptimal weighting scheme for given specific graph. Thus, one could think of computing and assigning optimal weights to the graph edges, such as to minimize the averaging time for the given graph topology. The problem of finding these optimal weighting factors can be formulated as a convex optimization problem, and has been solved in [9, 12]. It is also shown, see e.g. [14], that this problem can be solved in a distributed fashion, that is, without knowledge of the complete graph topology at a single node.

In the setting of a time-constant topology, the fastest mixing algorithm has been compared to different weighting schemes (e.g. uniform weights or Metropolis weights) in [9, 15]. This comparison showed that weight optimization can actually make a significant difference. [9] provides examples in which the optimized weights perform arbitrarily better than the Metropolis weighting algorithm.

This is, however, only in the case of a time-constant network, that is without link failures, or if the weights were optimized at each time step for the current topology. A simpler, yet suboptimal solution would be to optimize the weights for the case where all the links are up, and in the case of link failures, add the weight associated with the failing link to the self loops of the adjacent nodes. A comparison with the Metropolis–Hastings weighting scheme in [16] shows that this is advantageous only if the link failure probability is relatively small.

### 1.2.4 Gossip Algorithm

In gossip algorithms, each node communicates with no more than one neighbor in each time slot [17]. We make the difference between synchronized gossip and asynchronous gossip. In the synchronized case, at each time step, every node becomes active with a certain probability. Active nodes then choose randomly a neighbor to pair up with, and if the other node is available, both nodes connect and average their respective values. In the asynchronous model, we assume continuous instead of discrete time. Every node becomes active at an exponentially distributed random instant, connects with a random neighbor, and both nodes average their values. Thus, unlike in the synchronous case, communications occur sequentially, and never simultaneously. See [6] for a comprehensive treatment of gossip algorithms and their performance in terms of averaging speed, and [5] for a shorter version.

Gossiping is actually not a weighting scheme, but determines whether two nodes get to communicate at all on a given instant. However, if we identify communication links not activated by the

gossip algorithm in a given time slot with failing links, gossip is a special case of the time varying graph structure presented in the introduction. In the synchronous model, the nodes are restricted to have instantaneous degree one. The asynchronous model is even more restrictive, because there is exactly one active connection at every time step.

The weighting scheme typically used averages the two values of communicating nodes, that is

$$\begin{cases} w_{ij}(t) = w_{ii}(t) = 0.5 & \text{if nodes } i \text{ and } j \text{ are connected at time } t, \\ w_{kk}(t) = 1 & \text{if node } k \text{ has no connection at time } t, \\ w_{ij} = 0 & \text{on all other edges.} \end{cases} \tag{1.2.3}$$

This is technically the application of the Metropolis weights to the degenerate case where the instantaneous degree is at most one. As a possible generalization, one might also consider using a different weight on the link and changing the self-loop weights accordingly. Further, the choice of the communication pair $(i, j)$ is not trivial either, and may not be restricted to direct neighbors. Allowing remote nodes to exchange their values directly, using some kind of routing through the network, may significantly improve performance [7].

## 1.3 Notations and Definitions

### 1.3.1 Matrix-Vector Notation

In the sequel, we will use a matrix-vector notation, which will prove very convenient and will allow us to use known results, especially in the theory of random matrices. We first define the *state vector*

$$\mathbf{x}(t) := \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{pmatrix} \qquad t \geq 0 \tag{1.3.1}$$

as a function of $t$ by collecting the current values at nodes $1$ through $n$. The initial measurements are thus collected in the vector $\mathbf{x}(0)$. Likewise, we collect the weighting factors in a time-dependant *weighting matrix*

$$\mathbf{W}(t) := \begin{pmatrix} w_{11}(t) & w_{12}(t) & \cdots & w_{1n}(t) \\ w_{21}(t) & w_{22}(t) & & w_{2n}(t) \\ \vdots & & \ddots & \vdots \\ w_{n1}(t) & w_{n2}(t) & \cdots & w_{nn}(t) \end{pmatrix} \qquad t \geq 0. \tag{1.3.2}$$

The update equation (1.1.1) can then be rewritten as

$$\mathbf{x}(t+1) = \mathbf{W}(t)\mathbf{x}(t) \qquad t \geq 0, \tag{1.3.3}$$

and using it recursively we can write the current state vector $\mathbf{x}(t)$ as

$$\mathbf{x}(t) = \prod_{p=1}^{t} \mathbf{W}(t-p)\mathbf{x}(0). \tag{1.3.4}$$

The desired behavior for the sequence $\mathbf{x}(t)$ is to converge to the vector $\bar{x}\mathbf{1}$, where $\bar{x}$ is the average of the initial measurements, as defined in equation (1.1.2), and

$$\mathbf{1}_n := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \tag{1.3.5}$$

is the $n$-dimensional vector of all ones. $\mathbf{1}$ without subscript is used when the length of the vector is obvious from the context.

Our main focus will be on the error vector $\mathbf{y}(t)$ given by the difference

$$\mathbf{y}(t) := \mathbf{x}(t) - \bar{x}\mathbf{1} \tag{1.3.6}$$

and its rate of convergence to zero.

### 1.3.2 Mathematical Model

**Assumptions**

We will study the behavior of the vector-valued sequence of random variables $\{\mathbf{x}(t)\}_{t\geq 0}$, as defined in equation (1.3.3). The underlying random process is the one governing the sequence of matrices $\{\mathbf{W}(t)\}_{t\geq 0}$. Unless otherwise noted, we assume conditions (1.1.3) and (1.1.4) hold at all times, i.e.

$$\mathbf{W}(t)\mathbf{1} = \mathbf{1} \qquad \text{and} \tag{1.3.7}$$

$$\mathbf{1}^{\mathrm{T}}\mathbf{W}(t) = \mathbf{1}^{\mathrm{T}} \qquad \forall t. \tag{1.3.8}$$

Other than that, we make only minimal assumptions on the sequence $\{\mathbf{W}(t)\}$. Throughout the paper, we will assume it to be a stationary sequence, i.e. $\forall 0 \leq s \leq t$, the joint distribution of $\{\mathbf{W}(s), \mathbf{W}(s+1), \ldots, \mathbf{W}(t)\}$ is the same than that of $\{\mathbf{W}(s+1), \mathbf{W}(s+2), \ldots, \mathbf{W}(t+1)\}$.

Further assumptions such as ergodicity, of mutual independence of the matrices $\mathbf{W}(t)$ shall be specified accordingly. In particular, ergodicity of the random sequence largely simplifies the results presented in Chapter 3, and the results given in Chapter 4 require the matrices $\mathbf{W}(t)$ to be independent identically distributed (i.i.d.).

Finally, the most restrictive case is when the weighting matrix does not vary over time, $\mathbf{W}(t) = \mathbf{W}$ for all $t$. This particular case is briefly described in Section 3.1.

**Analogy with Markov Chains**

Without loss of generality, the initial measurement vector $\mathbf{x}(0)$ can be scaled and shifted so as to be non-negative and to satisfy $\sum x_i(0) = 1$.

If the weighting factors are all non-negative and less than or equal to one, the time-varying sensor network can be compared to a inhomogeneous Markov chain, the vectors $\mathbf{x}(t)$ being probability distributions on the chain, and the weighting factors $w_{ij}(t)$ transition probabilities between states of the chain. $\mathbf{W}(t)$ are therefore according transition probability matrices.

Consequently, we may sometimes refer to $\mathbf{W}$ as right, left, or doubly stochastic matrices, depending on the context. These terms denote non-negative matrices satisfying respectively condition (1.3.7), (1.3.8), or both.

However, as we do not actually deal with probability distributions, the non-negativity requirement will mostly not be crucial in our discussions. But even so, this requirement is not very stringent. Among the weighting schemes presented in Section 1.2, negative weights or weights greater than one may only occur in the optimized weights scheme (Paragraph 1.2.3).

### 1.3.3 Useful Functions

The following useful functions and convenient notations are used in the sequel. The suggestive notations

$$\mathbf{M} > \alpha \text{ and } \mathbf{M} \geq \alpha$$

signify all the elements of a matrix $\mathbf{M}$ are greater, respectively greater or equal to, a scalar $\alpha$.

For common matrix functions, the usual notation shall be used. $\mathbf{M}^{\mathrm{T}}$ denotes the transpose of a matrix $\mathbf{M}$, $\lambda_i(\mathbf{M})$ its eigenvalue with the $i$-th largest modulus, $\sigma_i(\mathbf{M})$ its $i$-th largest singular value, $\rho(\mathbf{M}) = |\lambda_1(\mathbf{M})|$ its spectral radius, and $\mathrm{tr}(\mathbf{M})$ its trace.

Furthermore, $\mathbf{A} \otimes \mathbf{B}$ denotes the *Kronecker product* of two matrices $\mathbf{A}$ and $\mathbf{B}$. The notation $\mathbf{e}_i$ will be used to denote the $i$-th standard basis vector, and $\mathbf{1}$ to denote the vector of all ones, as previously.

Finally, we define the function $\log^+(x)$ as

$$\log^+(x) := \max\left(0, \log(x)\right). \tag{1.3.9}$$

# 2. Convergence

## 2.1 Sufficient Conditions for Convergence

Our goal is to compute the average of all initial measurements at every node, that is to make all $x_i(t)$, $i = 1 \dots n$ converge to $\bar{x}$, or equivalently that $\mathbf{x}(t)$ goes to $\bar{x}\mathbf{1}_n$. In this section, we will establish sufficient conditions on the random process $\{\mathbf{W}(t)\}_{t\geq 0}$ for this to happen.

We first define the *stopping time*

$$T_\varepsilon := \inf_t \{t \geq 1 \,:\, \prod_{p=1}^{t} \mathbf{W}(t-p) \geq \varepsilon > 0\} \tag{2.1.1}$$

with respect to the random process $\{\mathbf{W}(t)\}_{t\geq 0}$, where $\varepsilon$ is an arbitrarily small positive number. The notation $\mathbf{A} \geq \varepsilon$ means that all the elements of the matrix $\mathbf{A}$ are greater than or equal to $\varepsilon$. With this definition, and the notation defined in Section 1.3, we may state the following theorem.

**Theorem 2.1.1.** *(Sufficient conditions for convergence) The sequence*

$$\mathbf{x}(t) = \prod_{p=1}^{t} \mathbf{W}(t-p)\mathbf{x}(0) \quad t \geq 1$$

*converges to $\bar{x}\mathbf{1}$ with probability 1 if the the following conditions hold:*
*1) The sequence $\{\mathbf{W}(t)\}_{t\geq 0}$ is stationary*
*2) $\mathbf{W}(t)\mathbf{1} = \mathbf{1}$ w.p. 1, $\forall t$ (fixed point)*
*3) $\mathbf{1}^{\mathrm{T}}\mathbf{W}(t) = \mathbf{1}^{\mathrm{T}}$ w.p. 1, $\forall t$ (preservation of sum)*
*4) $\|\mathbf{W}(t)\|_2 \leq 1$ w.p. 1, $\forall t$ (contracting property)*
*5) $\forall \varepsilon > 0,\ \mathrm{E}[T_\varepsilon] < \infty$*
*where $\mathrm{E}[T_\varepsilon] < \infty$ denotes the expectation of the stopping time $T_\varepsilon$ defined in equation (2.1.1).*

*Remark.* The inequality stated in Condition 4, together with either Condition 2 or 3 actually implies equality $\|\mathbf{W}(t)\|_2 = 1$.
*Remark.* Condition 5 relates to the connectivity of the network. In the model where communication links fail independently of each other, this is equivalent to having a connected graph.
*Remark.* This widely generalizes the convergence theorem in [18].

*Proof.* Outline of the proof: we will start by showing that maximum difference between elements of $\mathbf{x}(T_\varepsilon)$ is strictly less than between elements of $\mathbf{x}(0)$, then use arguments from renewal theory to prove that a subsequence of $\{\mathbf{x}(t)\}_{t\geq 0}$ converges a.s. Finally, we use Condition 4 to infer convergence of the whole sequence.

To simplify the notation, define $\mathbf{P} := \prod_{p=1}^{T_\varepsilon} \mathbf{W}(T_\varepsilon - p)$ and denote by $p_{ij}$ the i-th row, j-th column element of this matrix. Using Condition 2

$$\mathbf{W}(t)\mathbf{1} = \mathbf{1}\ \forall t \Rightarrow \mathbf{P}\mathbf{1} = \mathbf{1} \Leftrightarrow \sum_j p_{ij} = 1\ \forall i = 1 \dots n$$

and the definition of $T_\varepsilon$, we can bound the maximum element of $\mathbf{x}(T_\varepsilon)$:

$$\begin{aligned}\max_i x_i(T_\varepsilon) &= \max_i \sum_j p_{ij} x_j(0) \\ &\leq (1-\varepsilon) \max_j x_j(0) + \varepsilon \min_j x_j(0).\end{aligned}$$

Similarly, one can bound the minimum element from below:

$$\min_i x_i(T_\varepsilon) \geq \varepsilon \max_j x_j(0) + (1-\varepsilon) \min_j x_j(0).$$

Taking the difference of the two yields

$$\begin{aligned}\max_i x_i(T_\varepsilon) - \min_i x_i(T_\varepsilon) \leq & (1-\varepsilon) \max_j x_j(0) + \varepsilon \min_j x_j(0) \\ & - \varepsilon \max_j x_j(0) - (1-\varepsilon) \min_j x_j(0) \\ = & (1-2\varepsilon) \max_j x_j(0) - (1-2\varepsilon) \min_j x_j(0) \\ = & (1-2\varepsilon) \left( \max_j x_j(0) - \min_j x_j(0) \right).\end{aligned} \tag{2.1.2}$$

Now define an increasing sequence of integers $\{k^{(l)}\}_{l\geq 0}$ as follows:

$$k^{(0)} = 0,\ k^{(1)} = T_\varepsilon, \ldots,\ k^{(l+1)} = \inf\{t \geq k^{(l)}+1 : \prod_{p=1}^{t-k^{(l)}} \mathbf{W}(t-p) \geq \varepsilon\}, \tag{2.1.3}$$

and an associated counting process

$$N(t) := \max\ l \quad \text{s.t. } k^{(l)} \leq t, \qquad \forall t \geq 0. \tag{2.1.4}$$

That is, we group the matrices $\mathbf{W}(t)$ of the sequence in such a way to form a sequence of positive matrices. The counter $N(t)$ denotes the number of such groups up to time $t$.

As the process $\{\mathbf{W}(t)\}_{t\geq 0}$ is invariant under arbitrary time shifts, the time intervals $k^{(l+1)} - k^{(l)}$ are i.i.d. for all $l \geq 0$. The process $\{N(t), t \geq 0\}$ is therefore a *renewal process* [13] with inter-renewal time $T_\varepsilon$.

By the *strong law for renewal processes* [13, 19], we know

$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{\mathrm{E}[T_\varepsilon]} \quad \text{w.p.1.} \tag{2.1.5}$$

Now get back to the sequence $\mathbf{x}(t)$, and iterate equation (2.1.2) $l$ times:

$$\max_i x_i\left(k^{(l)}\right) - \min_i x_i\left(k^{(l)}\right) \leq (1-2\varepsilon)^l \left( \max_j x_j(0) - \min_j x_j(0) \right), \tag{2.1.6}$$

and consider the modified sequence $\{\hat{\mathbf{x}}(t)\}_{t\geq 0}$ obtained by sampling the original process at the renewal times, i.e. defined by

$$\hat{\mathbf{x}}(t) := \mathbf{x}\left(k^{(N(t))}\right). \tag{2.1.7}$$

This modified sequence allows us to rewrite equation (2.1.6) in terms of $t$

$$\max_i \hat{x}_i(t) - \min_i \hat{x}_i(t) \leq (1-2\varepsilon)^{N(t)} \left( \max_j \hat{x}_j(0) - \min_j \hat{x}_j(0) \right). \tag{2.1.8}$$

Because of Condition 5 and of (2.1.5), the right hand side of (2.1.8) tends to 0 for large $t$, hence the left hand side of (2.1.8) tends to 0 for large $t$ as well, and therefore

$$\lim_{t\to\infty} \hat{\mathbf{x}}(t) = c \cdot \mathbf{1} \quad \text{w.p.1.} \tag{2.1.9}$$

By the preservation of sum of the elements of $\mathbf{x}(t)$ (condition 3), the constant $c$ must be equal to $\bar{x}$.

It now remains to show that convergence of $\{\hat{\mathbf{x}}(t)\}_{t\geq 0}$ implies convergence of $\{\mathbf{x}(t)\}_{t\geq 0}$. For that, note that we can always rewrite

$$\mathbf{x}(t) = \mathbf{y}(t) + \bar{x}\mathbf{1}, \tag{2.1.10}$$

where the second term is invariant and the first term $\mathbf{y}(t)$ has the sum of its elements equal to zero. That is we may, without loss of generality, consider only sequences of the type $\mathbf{y}(t)$ and show such sequences go to zero.

By equation (2.1.9), we know that a sequence $\hat{\mathbf{y}}(t) := \hat{\mathbf{x}}(t) - \bar{x}\mathbf{1}$ vanishes. But, by condition 4, we know

$$\begin{aligned} \|\mathbf{y}(t+1)\|_2 =& \|\mathbf{W}(t)\mathbf{y}(t)\|_2 \\ \leq& \|\mathbf{W}(t)\|_2 \|\mathbf{y}(t)\|_2 \\ \leq& \|\mathbf{y}(t)\|_2 \\ \leq& \left\|\mathbf{y}\left(k^{(N(t))}\right)\right\|_2 \\ =& \|\hat{\mathbf{y}}(t)\|_2, \end{aligned} \tag{2.1.11}$$

i.e. the sequence $\{\|\hat{\mathbf{y}}(t)\|_2\}_{t\geq 0}$ bounds from above the sequence $\{\|\mathbf{y}(t+1)\|_2\}_{t\geq 0}$. Therefore,

$$\lim_{t\to\infty} \|\mathbf{y}(t)\|_2 = 0 \tag{2.1.12}$$

or equivalently, with (2.1.10)

$$\lim_{t\to\infty} \mathbf{x}(t) = \bar{x}\mathbf{1}. \tag{2.1.13}$$

□

## 2.2 Exponential Convergence

The *strong law for renewal processes* (2.1.5) actually allows a stronger result than Theorem 2.1.1. The careful reader may already have noticed that the decay rate of the quantities on both sides of equation (2.1.8) is exponential, and that the rate of this exponential decay may be expressed in terms of $\varepsilon$. This leads us to the following result.

**Theorem 2.2.1.** *(A first bound on the exponential rate of convergence)*

$$\lim_{t\to\infty} \frac{1}{t} \log \left( \frac{\max_i x_i(t) - \min_i x_i(t)}{\max_i x_i(0) - \min_i x_i(0)} \right) \leq \inf_{0<\varepsilon<1/n} \frac{1}{\mathrm{E}[T_\varepsilon]} \log(1-2\varepsilon)$$

*with $T_\varepsilon$ defined as in equation (2.1.1).*

*Remark.* $T_\varepsilon$ depends on $\varepsilon$. Small $\varepsilon$ lead to small $\mathrm{E}[T_\varepsilon]$ and vice-versa. Only after minimization over $\varepsilon$, one is assured to have the tightest possible bound with this method.

*Proof.* To simplify the notation, define the function

$$d : \mathbb{R}^n \to \mathbb{R}_0^+ \; : \; \mathbf{x} \mapsto d(\mathbf{x}) := \max_i x_i - \min_i x_i. \tag{2.2.1}$$

Recall equation (2.1.8),

$$\frac{d(\hat{\mathbf{x}}(t))}{d(\hat{\mathbf{x}}(0))} \leq (1-2\varepsilon)^{N(t)} \Leftrightarrow \log \frac{d(\hat{\mathbf{x}}(t))}{d(\mathbf{x}(0))} \leq N(t) \log(1-2\varepsilon) \tag{2.2.2}$$

and note that inequality (2.1.11) implies $d(\mathbf{x}(t))/d(\hat{\mathbf{x}}(t))$ is bounded (refer to Lemma 2.2.2 for details). Then, a few simple calculations and the use of the strong law for renewal processes lead to the desired result.

$$\begin{aligned}
\lim_{t\to\infty} \frac{1}{t} \log \frac{d(\mathbf{x}(t))}{d(\mathbf{x}(0))} &= \lim_{t\to\infty} \frac{1}{t} \log \frac{d(\mathbf{x}(t))d(\hat{\mathbf{x}}(t))}{d(\hat{\mathbf{x}}(t))d(\mathbf{x}(0))} && (2.2.3)\\
&= \lim_{t\to\infty} \frac{1}{t} \log \frac{d(\mathbf{x}(t))}{d(\hat{\mathbf{x}}(t))} + \lim_{t\to\infty} \frac{N(t)}{t} \frac{1}{N(t)} \log \frac{d(\hat{\mathbf{x}}(t))}{d(\mathbf{x}(0))} && (2.2.4)\\
&= 0 + \lim_{t\to\infty} \frac{N(t)}{t} \cdot \lim_{t\to\infty} \frac{1}{N(t)} \log \frac{d(\hat{\mathbf{x}}(t))}{d(\mathbf{x}(0))} && (2.2.5)\\
&= \frac{1}{\mathrm{E}[T_\varepsilon]} \cdot \lim_{t\to\infty} \frac{1}{N(t)} \log \frac{d(\hat{\mathbf{x}}(t))}{d(\mathbf{x}(0))} && (2.2.6)\\
&\leq \frac{1}{\mathrm{E}[T_\varepsilon]} \log (1-2\varepsilon) && (2.2.7)
\end{aligned}$$

□

**Lemma 2.2.2.**

$$\frac{\|\mathbf{y}(t)\|_2}{\|\hat{\mathbf{y}}(t)\|_2} \leq 1 \Rightarrow \frac{d(\mathbf{x}(t))}{d(\hat{\mathbf{x}}(t))} \leq \frac{\sqrt{n}}{2}$$

*Proof.* The result follows readily from the following two inequalities:

$$\|\mathbf{y}(t)\|_2 \geq \frac{1}{\sqrt{2}} d(\mathbf{x}(t)) \tag{2.2.8}$$

$$\|\hat{\mathbf{y}}(t)\|_2 \leq \sqrt{\frac{n}{2}} d(\hat{\mathbf{x}}(t)). \tag{2.2.9}$$

For the derivation of these, we will use $y_M$ and $y_m$ to denote the largest and the smallest elements of $\mathbf{y}(t)$, with $y_M = \max(y_M, y_m)$ and $y_m = \min(y_M, y_m)$. Note that $y_M$ and $y_m$ have opposite signs, because the sum of all elements of $\mathbf{y}(t)$ equals zero. For the first inequality, first note that

$$\begin{aligned}
y_M^2 + y_m^2 &= [(y_M + y_m) - y_m]^2 + y_m^2 \\
&= (y_M + y_m)^2 - 2y_M y_m \\
&\geq -2y_M y_m. && (2.2.10)
\end{aligned}$$

Then, inequality (2.2.8) may be derived by bounding $\|\mathbf{y}(t)\|_2^2$ as follows:

$$\begin{aligned}\|\mathbf{y}(t)\|_2^2 = \sum_{i=1}^{n} y_i^2 &\geq y_M^2 + y_m^2 \\ &\geq \frac{1}{2}\left(y_M^2 + y_m^2\right) - y_M y_m \\ &= \frac{1}{2}\left(y_M - y_m\right)^2 = \frac{1}{2}d^2\left(\mathbf{y}\left(t\right)\right) = \frac{1}{2}d^2\left(\mathbf{x}\left(t\right)\right).\end{aligned}$$

To derive inequality (2.2.9), first note that $\sum_i y_i = 0$ implies that at least half of the elements of $\mathbf{y}(t)$ have magnitude less than or equal to $|y_m|$. Consequently,

$$\begin{aligned}\|\mathbf{y}(t)\|_2^2 = \sum_{i=1}^{n} y_i^2 &\leq \frac{n}{2}y_M^2 + \frac{n}{2}y_m^2 \\ &\leq \frac{n}{2}\left(y_M - y_m\right)^2 = \frac{n}{2}d^2\left(\mathbf{y}\left(t\right)\right) = \frac{n}{2}d^2\left(\mathbf{x}\left(t\right)\right).\end{aligned}$$

□

The bound derived in this section might be useful in estimating the convergence speed of an averaging algorithm, but still has two major disadvantages. First, the quantity $\mathrm{E}[T_\varepsilon]$ is in general not easy to compute, even for a fixed $\varepsilon$. And more importantly, this bound is not guaranteed to be reasonably tight, even after optimization over $\varepsilon$[1]. In Chapter 4, we will derive other bounds on the asymptotic contraction rate, but those hold only in the case of i.i.d. weighting matrices.

Finally, note that this bound is based on the quantity $d(\mathbf{x}(t))$ which is a semi-norm whose kernel is the subspace tangent to the $\mathbf{1}_n$ vector. This fact may help to relate $d(\mathbf{x}(t))$ with Dobrushin's ergodic coefficient, a matrix seminorm we will introduce in Section 3.2.2.

[1]Further investigation in this direction is not pursued herein, but might be advisable in order to clarify the practicability of this bound.

# 3. Products of Random Matrices

In this chapter, we take a step back from the averaging problem in time-varying networks presented in the previous chapters, to look at a broader picture. We study the behavior of infinite products of random matrices, with only minimal assumptions on the distribution of the matrices. That is, for a given sequence of matrices $\{\mathbf{A}(t), t = 0, 1, 2, \ldots\}$, we investigate the product sequence

$$\mathbf{M}_t := \prod_{p=1}^{t} \mathbf{A}(t-p) \qquad t = 1, 2, 3, \ldots$$

Our main assumption will be that the sequence $\{\mathbf{A}(t)\}_{t\geq 0}$ is *stationary*, which will allow us to use results from ergodic theory. For more specific results, we will assume the matrices to be *ergodic*, or even *i.i.d.* This setting applies to the the special case of the averaging problem, as it becomes clear in Section 3.2.2, but has much broader applications in general. These range from demography analysis [20] to topics in theoretical physics, such as the study of directed polymers in random environments [21].

While the first section of this chapter is meant as an introductory example, Sections 3.2 and 3.3 are mainly a review on subadditive ergodic theory, and most of the results presented are well known to those familiar with the subject. Theorem 3.2.4 is the central result of subadditive ergodic theory [22, 23, 24, 25, 26]. The following Theorem 3.2.8 is technically a Corollary of the latter, but it was presented by Fuerstenberg and Kesten in [27] before subadditive ergodic theory was even invented, and deserves particular attention because of its focus on the random matrix product application. Section 3.3 presents a theorem that describes the whole spectrum of an infinite matrix product, rather than the largest eigenvalue only. It was first presented by Oseledec in [28], and other proofs were later given by Ruelle [29] and Raghunathan [30]. A very good overview on all these results is given by Walters at the end of his book on ergodic theory [26]. We conclude this review by presenting an alternative point of view on the main results discussed (Section 3.4), and by applying them to the special case of the averaging problem.

## 3.1 Infinite Products of Constant Matrices

First consider the simple case where the elements in the sequence $\{\mathbf{A}(t)\}$ are all equal and non-random, that is $\mathbf{A}(t) = \mathbf{A}$ for all $t$. In this case, the product sequence $\{\mathbf{M}_t\}_{t\geq 1}$ is given by

$$\mathbf{M}_t = \mathbf{A}^t \qquad t = 1, 2, 3, \ldots$$

The matrix $\mathbf{A}$ can be expressed using its Jordan canonical form [31], $\mathbf{A} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$, where $\mathbf{V}$ contains the eigenvectors of $\mathbf{A}$ and $\mathbf{J}$ is block diagonal,

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_k \end{pmatrix}$$

with Jordan blocks

$$\mathbf{J}_i = \begin{pmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{pmatrix}.$$

Consequently, $\mathbf{M}_t = \mathbf{A}^t = \mathbf{V}\mathbf{J}^t\mathbf{V}^{-1}$. It is useful to factor out the spectral radius $\rho = |\lambda_1(\mathbf{A})|$ of $\mathbf{A}$, which leads to

$$\mathbf{M}_t = \rho^t \cdot \mathbf{V}\rho^{-t}\mathbf{J}^t\mathbf{V}^{-1}. \tag{3.1.1}$$

For large $t$, all Jordan blocks $\rho^{-t}\mathbf{J}_i^t$ in $\rho^{-t}\mathbf{J}^t$ with $|\lambda_i| < \rho$ will vanish, and the matrix $\mathbf{M}_t$ presents one of the following three behaviors, depending on the spectral radius $\rho(\mathbf{A})$:

$$\begin{cases} \rho(\mathbf{A}) < 1 & \Rightarrow \quad \mathbf{M}_t \to 0 \\ \rho(\mathbf{A}) = 1 & \Rightarrow \quad \mathbf{M}_t \to \mathbf{M} \neq 0 \\ \rho(\mathbf{A}) > 1 & \Rightarrow \quad \mathbf{M}_t \text{ diverges.} \end{cases}$$

The spectral radius actually determines the contraction ($\rho < 1$) or expansion ($\rho > 1$) rate of a system governed by the sequence $\{\mathbf{A}(t)\}$, as is shown by the following formula.

**Theorem 3.1.1.** *For any matrix norm $\|\bullet\|$ (see Definition 3.2.7),*

$$\lim_{t\to\infty} \left\|\mathbf{A}^t\right\|^{1/t} = \rho(\mathbf{A}). \tag{3.1.2}$$

*Proof.* We refer to the book "Matrix Analysis" by Horn and Johnson [31, page 299] for the proof. □

If $\rho(\mathbf{A}) \neq 0$, we can rewrite this as

$$\lim_{t\to\infty} \frac{1}{t} \log \left\|\mathbf{A}^t\right\| = \log \rho(\mathbf{A}). \tag{3.1.3}$$

So in the case where all matrices in the sequence are the same, the contraction (or expansion) rate of the sequence $\{\mathbf{A}(t)\}$ is given by the spectral radius $\rho(\mathbf{A})$ of the matrix $\mathbf{A}$. The further $\rho$ is away from 1, the faster the contraction (or expansion) will be.

The question now is, how does this extend to the case where $\mathbf{A}(t)$ varies over time? Is there any way to specify or to compute an asymptotic contraction rate

$$\lim_{t\to\infty} \frac{1}{t} \log \left\| \prod_{p=1}^{t} \mathbf{A}(t-p) \right\|, \tag{3.1.4}$$

and when is this quantity well-defined in the first place? This is what we are going to investigate in the remainder of this chapter.

## 3.2 Kingman's Subadditive Ergodic Theorem

Consider a sequence of matrices $\{\mathbf{A}(t)\}_{t\geq 0}$ with some joint distribution function. We are interested in the asymptotic behavior of the product sequence

$$\mathbf{M}_t := \prod_{p=1}^{t} \mathbf{A}(t-p) \qquad t = 1, 2, 3, \ldots \tag{3.2.1}$$

In contrast to the constant case presented before, it is already not trivial to assert whether quantity (3.1.4) is well defined. Moreover, this quantity is non-random only in some special cases. This result is essentially what Kingman's Subadditive Ergodic Theorem says (Theorem 3.2.4, [23]).

The following paragraphs retrace parts of the path followed by Cohen in [25], which shows how the subadditive ergodic theorem simultaneously extends an elementary theorem of analysis (Theorem 3.2.1) and Birkhoff's ergodic theorem (Theorem 3.2.2). Important details and remarks omitted therein were drawn directly from Kingman's work [22, 23].

**Theorem 3.2.1.** *(Subadditive Theorem) Let* $\{a_n;\, n = 1, 2, 3, \ldots\}$ *be a sequence of real numbers such that*

$$a_{m+n} \leq a_m + a_n \qquad m, n = 1, 2, 3, \ldots. \tag{3.2.2}$$

*Then, the sequence* $\{a_n/n,\, n = 1, 2, 3, \ldots\}$ *either converges to its lower bound, or diverges properly to* $-\infty$.

Note that if we replace the hypothesis of subadditivity (3.2.2) by that of additivity, $a_{m+n} = a_m + a_n$, we have trivially $a_n/n = a_1$ for all $n$.

*Proof.* Let $\gamma := \inf_{n\geq 1} a_n/n$. If $\gamma = -\infty$, there is nothing to prove. If $\gamma > -\infty$, for any arbitrary $\varepsilon > 0$, one can find an $m$ such that $a_m/m < \gamma + \varepsilon$.

Given any integer $m$, an integer $n$ can be written as $n = qm + r$, with $q$ being integer, and $0 \leq r \leq m - 1$. Define $a_0 = 0$. Subadditivity implies

$$a_n = a_{mq+r} \leq a_m + \ldots (q \text{ times}) \ldots + a_m + a_r = qa_m + a_r. \tag{3.2.3}$$

Further,

$$\gamma \leq \frac{a_n}{n} \leq \frac{qa_m + a_r}{n} = \frac{qa_m}{n} + \frac{a_r}{n} \tag{3.2.4}$$

$$\Rightarrow \gamma \leq \frac{a_n}{n} \leq \frac{qm}{n}\frac{a_m}{m} + \frac{a_r}{n} < \frac{qm}{n}(\gamma + \varepsilon) + \frac{a_r}{n} \tag{3.2.5}$$

If we let $n \to \infty$, we have $qm/n \to 1$ and $a_r/n \to 0$. Since $\varepsilon$ is arbitrarily close to zero, $a_n/n \to \gamma$. □

**Theorem 3.2.2.** *(Birkhoff's Ergodic Theorem) Let* $\{y_n;\, n = 1, 2, 3, \ldots\}$ *be a stationary sequence of random variables with finite expectation. Then the limit*

$$\eta = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} y_j \tag{3.2.6}$$

*exists with probability one and in expectation, and*

$$\mathrm{E}[\eta] = \mathrm{E}[y_1]. \tag{3.2.7}$$

A proof of the Birkhoff - Von Neumann Theorem can be found e.g. in [23], or in any book on ergodic theory, for example in [26].

The two previous theorems seem quite different. While Theorem 3.2.1 deals with *deterministic subadditive* sequences, Theorem 3.2.2 considers *random additive* sequences. However, the two can be merged into the following theorem, which considers the limiting behavior *random subadditive* sequences. Let us first introduce a definition.

**Definition 3.2.3.** (Subadditive Ergodic Process, SEP) A SEP is a family of random variables $\mathcal{X} := \{x_{s,t},\ s,t \in T,\ s < t\}$, $T$ being the set of non-negative integers, satisfying the following conditions:

**(S1)** $x_{s,u} \leq x_{s,t} + x_{t,u} \quad \forall\, s < t < u, \quad s,t,u \in T$ (subadditivity)

**(S2)** The joint distributions of the process $(x_{s+1,t+1})$ are the same as those of $(x_{s,t})$ (stationarity)

**(S3)** The expectation $g_t = \mathrm{E}\left[x_{0,t}\right]$ exists and satisfies $g_t \geq Ct$ for some constant C.

Condition (S3) implies that the considered limit will be finite. However, for some applications we may want to allow the limit to diverge to $-\infty$, in which case we can relax condition (S3) to

**(S3')** $\mathrm{E}\left[\max\left(x_{0,1},\ 0\right)\right] < \infty$.

With these preliminaries, we can now state Kingman's Theorem.

**Theorem 3.2.4.** *(Subadditive Ergodic Theorem) If $\mathcal{X}$ is a SEP, then the finite limit*

$$\xi = \lim_{t\to\infty} \frac{x_{0,t}}{t} \tag{3.2.8}$$

*exists with probability one and in mean, and*

$$\gamma := \mathrm{E}[\xi] = \inf_{t\geq 1} \frac{\mathrm{E}\left[x_{0,t}\right]}{t} = \lim_{t\to\infty} \frac{\mathrm{E}\left[x_{0,t}\right]}{t}. \tag{3.2.9}$$

*Moreover, if $\mathcal{X}$ satisfies (S1), (S2) and (S3') but not (S3), then the limit* (3.2.8) *exists with probability one in $-\infty \leq \xi < \infty$, and*

$$\mathrm{E}[\xi] = -\infty. \tag{3.2.10}$$

We do not reproduce the proof here. Proofs may be found in [22, 23, 24] or in [26].

In general, the random variable $\xi$ may be non-degenerate. However, if the random process $\mathbf{X} := \{x_{0,t}\}_{t\geq 0}$ is *ergodic*, $\xi = \gamma$ with probability one. To see what this means, recall the definition of ergodicity.

**Definition 3.2.5.** (Ergodicity) Let $(X, \mathcal{B}, \mu)$ be a probability space. A measure preserving transformation $T$ of $(X, \mathcal{B}, \mu)$ is called ergodic if the only events $B \in \mathcal{B}$ with $T^{-1}B = B$ satisfy $\mu(B) = 0$ or $\mu(B) = 1$.

If $X := \{x_t\}$ is a stochastic process, when we say $X$ is ergodic, what we really mean is that the time shift $T : X \to X' : x'_t := x_{t+1}$ is an ergodic transformation.

Going back to the SEP $\mathcal{X}$, if $\mathcal{F}$ denotes the $\sigma$-field of events defined in terms of $\mathcal{X}$ and invariant under the time shift $T : \mathcal{X} \to \mathcal{X}' : x'_{s,t} := x_{s+1,t+1}$, then $\xi$ is $\mathcal{F}$-measurable, with the explicit notation

$$\xi = \lim_{t\to\infty} \frac{1}{t} \mathrm{E}\left[x_{0,t}|\mathcal{F}\right]. \tag{3.2.11}$$

Consequently, if $\mathcal{F}$ is a trivial $\sigma$-field consisting only of events of probability zero or one, i.e. $\mathcal{X}$ is ergodic, $\xi = \gamma$ almost surely.

In some cases, $\mathcal{F}$ can be shown to be trivial using a zero-one law. An example where this is the case is when the processes are defined by equations of the form

$$x_{s,t} = F_{t-s}\left(\ldots, u_{s-1}, u_s, u_{s+1}, \ldots\right) \tag{3.2.12}$$

where the functions $F$ are fixed, and $u_s$ are i.i.d. random variables. The $\sigma$-field $\mathcal{F}$ is contained in the $\sigma$-field $\mathcal{F}'$ of events defined in terms of the $u_s$ and invariant under the time shift $u_s \to u_{s+1}$. $\mathcal{F}'$ is known to be trivial [32]. As $\mathcal{F} \subseteq \mathcal{F}'$, this implies $\mathcal{F}$ is also trivial and therefore $\xi = \gamma$ with probability one.

### 3.2.1 Application to Matrix Products

Theorem 3.2.4 is formulated in terms of a subadditive process, consisting of real random variables. In this section, we describe how to apply it to products of random matrices. For that, first specify appropriate functions from the space of matrices to the reals, which permit us to apply Theorem 3.2.4.

If $f : \mathbb{C}^{n\times n} \to \mathbb{R} : \mathbf{A} \mapsto f(\mathbf{A})$ denotes such a function, the key property it must satisfy is
**(P1)** $f(\mathbf{AB}) \leq f(\mathbf{A}) + f(\mathbf{B})$.

This is satisfied for example if $f(\mathbf{A}) = \log \|\mathbf{A}\|$, where $\|\bullet\|$ denotes any matrix seminorm with the submultiplicative property, as defined in [31] and below.

**Definition 3.2.6.** A *matrix seminorm* $\|\bullet\|$ is a function from $\mathbb{C}^{n\times n}$ to $\mathbb{R}$ satisfying

**(C1)** $\|\mathbf{A}\| \geq 0$ (nonnegative)
**(C2)** $\|\alpha\mathbf{A}\| = |\alpha| \, \|\mathbf{A}\| \quad \forall \alpha \in \mathbb{C}$ (homogeneous)
**(C3)** $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ (triangle inequality)
**(C4)** $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \, \|\mathbf{B}\|$ (submultiplicative)

**Definition 3.2.7.** A *matrix norm* is a matrix seminorm with the additional property

**(C1a)** $\|\mathbf{A}\| = 0 \Leftrightarrow \mathbf{A} = 0$ (positive)

*Remark.* In some books, e.g. [33], the definitions of norm and seminorm do not include the submultiplicative property (C4). However, this property is crucial to us. It is actually the only one we really need, together with (C1) for the logarithm to be well-defined.

**Theorem 3.2.8.** *Let $S = (V, \cdot)$ be a semigroup, and $\|\bullet\| : V \to \mathbb{R}^+$ a submultiplicative function. Let $\{\mathbf{A}(t)\}_{t\geq 0}$ be a stationary sequence of random elements of $V$ and suppose that*

$$\mathrm{E}\left[\log^+ \|\mathbf{A}(0)\|\right] < \infty. \tag{3.2.13}$$

*Then, if $\mathbf{M}_t$ is defined as in equation* (3.2.1)*, the limit*

$$\xi = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\| \qquad \textit{exists, and} \tag{3.2.14}$$

$$\gamma := \mathrm{E}[\xi] = \lim_{t\to\infty} \frac{1}{t} \mathrm{E}\left[\log \|\mathbf{M}_t\|\right]. \tag{3.2.15}$$

*Moreover, if the sequence $\{\mathbf{A}(t)\}_{t\geq 0}$ is ergodic,*

$$\xi = \gamma \qquad \textit{with probability one.} \tag{3.2.16}$$

**Corollary 3.2.9.** *In Theorem 3.2.8, if $\{\mathbf{A}(t)\}_{t\geq 0}$ is a sequence of i.i.d. elements, then $\{\mathbf{A}(t)\}_{t\geq 0}$ is ergodic, and therefore*

$$\xi = \gamma \qquad \textit{with probability one.} \tag{3.2.17}$$

*Proof.* Observe that

$$x_{s,t} = \log \|\mathbf{M}_{s,t}\| = \log \left\| \prod_{p=1}^{t-s} \mathbf{A}(t-p) \right\| \tag{3.2.18}$$

satisfies conditions (S1), (S2) and (S3'). The first part of the theorem then follows directly from Theorem 3.2.4.

For the second part of the theorem and for the Corollary, refer to the discussion following Theorem 3.2.4. More specifically, set $u_s = \mathbf{A}(s)$ in equation (3.2.12) of that paragraph. □

As one might already have guessed, an important special case of Theorem 3.2.8 is when the semigroup $S$ is the set of $n \times n$ matrices with standard matrix multiplication, and the function $\|\bullet\|$ is a norm on this set. In this case, we can show that the particular choice of norm does not affect the values of $\xi$ or $\gamma$.

**Theorem 3.2.10.** *If the semigroup $S$ is the space $V$ of real or complex $n \times n$ matrices with standard matrix multiplication, and $\|\bullet\|$ is a norm on $V$, then the limits* (3.2.14) *and* (3.2.15) *do not depend on the choice of norm in $V$.*

*Proof.* As $n$ is finite, all norms on $V$ are equivalent. Thus, for any two norms $\|\bullet\|_\alpha$ and $\|\bullet\|_\beta$ on $V$, there exist constants $C_m$ and $C_M$ such that $C_m \|\mathbf{A}\|_\alpha \leq \|\mathbf{A}\|_\beta \leq C_M \|\mathbf{A}\|_\alpha$ for all $\mathbf{A} \in V$. See e.g. [31, Chapter 5] for more details. Consequently,

$$\xi_\beta = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\|_\beta \geq \lim_{t\to\infty} \frac{1}{t} \log \left(C_m \|\mathbf{M}_t\|_\alpha\right) = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\|_\alpha = \xi_\alpha \tag{3.2.19}$$

$$\xi_\beta = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\|_\beta \leq \lim_{t\to\infty} \frac{1}{t} \log \left(C_M \|\mathbf{M}_t\|_\alpha\right) = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\|_\alpha = \xi_\alpha \tag{3.2.20}$$

which implies $\xi_\alpha = \xi_\beta$. A similar argument holds for $\gamma$. □

### 3.2.2 Special Case: Stochastic Matrices

We discuss the application of Theorem 3.2.8 to products of stochastic matrices [24]. A commonly used parameter for describing the "scrambling" property of a stochastic matrix $\mathbf{W}$ is its ergodicity coefficient $\hat{\delta}(\mathbf{W})$.

**Definition 3.2.11.** Dobrushin's ergodic coefficient

$$\hat{\delta}(\mathbf{W}) := \frac{1}{2} \max_{i,j} \sum_k |w_{ik} - w_{jk}|$$

As usual, $w_{ij}$ denotes the $i$-th row $j$-th column element of $\mathbf{W}$. It is easy to see that if $\mathbf{W}$ is right stochastic, $0 \leq \hat{\delta}(\mathbf{W}) \leq 1$, where $\hat{\delta}(\mathbf{W}) = 1$ iff $\mathbf{W}$ has two orthogonal rows. Note however that the definition extends to any real (and even complex) matrix.

In some sense, the parameter $\hat{\delta}$ measures how far a stochastic matrix $\mathbf{W}$ is from a perfectly scrambling matrix $\mathbf{1}\boldsymbol{\pi}^{\mathrm{T}}$ which maps any probability distribution vector $\mathbf{x}^{\mathrm{T}}$ to the stationary distribution $\boldsymbol{\pi}^{\mathrm{T}}$.

Due to the inhabitual convention of using row vectors in Markov chain theory, Dobrushin's coefficient is fitted to *right* stochastic matrices (with *row* sums = 1). However, because we adopted a

column vector notation, we consider *left* stochastic matrices, with *column* sums equal to 1. Therefore, we will use the "transposed" version

$$\delta(\mathbf{W}) := \hat{\delta}(\mathbf{W}^{\mathrm{T}}) = \frac{1}{2}\max_{k,l}\sum_{i}|w_{ik}-w_{il}|\,. \tag{3.2.21}$$

As the function $\delta(\mathbf{W})$ is a seminorm, $\log(\delta(\bullet))$ satisfies the conditions of Theorem 3.2.8. Consequently, if $\{\mathbf{W}(t)\}_{t\geq 0}$ is a stationary sequence of (not necessarily stochastic!) matrices,

$$\xi = \lim_{t\to\infty}\frac{1}{t}\log\delta\left(\prod_{p=1}^{t}\mathbf{W}(t-p)\right) \tag{3.2.22}$$

exists with probability one and in expectation, and

$$\gamma = \mathrm{E}[\xi] = \lim_{t\to\infty}\frac{1}{t}\,\mathrm{E}\left[\log\delta\left(\prod_{p=1}^{t}\mathbf{W}(t-p)\right)\right]. \tag{3.2.23}$$

Now consider the case where all $\mathbf{W}$ are stochastic and have a common stationary distribution $\boldsymbol{\pi}$. Assume further they are diagonalizable.[1] The eigenvalue decomposition of $\mathbf{W}$ then yields

$$\mathbf{W} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1} = \left(\begin{array}{c:cc}\boldsymbol{\pi} & & \\ & * & \\ & & \end{array}\right)\begin{pmatrix}1 & & 0\\ & * & \\ 0 & & \ddots\end{pmatrix}\left(\begin{array}{c}\mathbf{1}_n^{\mathrm{T}}\\ \hdashline * \\ \end{array}\right) \tag{3.2.24}$$

$$= \boldsymbol{\pi}\mathbf{1}_n^{\mathrm{T}} + \mathbf{A} \tag{3.2.25}$$

where $\mathbf{A}$ satisfies $\mathbf{A}\boldsymbol{\pi} = 0$ and $\mathbf{1}_n^{\mathrm{T}}\mathbf{A} = 0$.

**Proposition 3.2.12.** *$\delta(\bullet)$ is a norm on the set of $\mathbf{A}$ as defined above.*

*Proof.* $\delta(\bullet)$ is obviously a seminorm, so the only thing we need to prove is $\delta(\mathbf{A}) = 0 \Rightarrow \mathbf{A} = 0$. By the triangle inequality,

$$\begin{aligned}\delta(\mathbf{W}) &\leq \delta(\boldsymbol{\pi}\mathbf{1}_n^{\mathrm{T}}) + \delta(\mathbf{A}) = \delta(\mathbf{A})\\ \delta(\mathbf{A}) &\leq \delta(-\boldsymbol{\pi}\mathbf{1}_n^{\mathrm{T}}) + \delta(\mathbf{W}) = \delta(\mathbf{W})\\ \Rightarrow \delta(\mathbf{A}) &= \delta(\mathbf{W}).\end{aligned}$$

Thus, $\delta(\mathbf{A}) = 0 \Rightarrow \delta(\mathbf{W}) = 0 \Leftrightarrow \exists\mathbf{p}$ s.t. $\mathbf{W} = \mathbf{p}\mathbf{1}_n^{\mathrm{T}}$. If $\mathbf{W} = \mathbf{p}\mathbf{1}_n^{\mathrm{T}}$, $\mathbf{p}$ must be the only stationary distribution of $\mathbf{W}$. Consequently, $\mathbf{p} = \boldsymbol{\pi}$ and $\mathbf{A} = 0$. □

Therefore, to study the product of a sequence of stochastic matrices $\{\mathbf{W}(t)\}_{t\geq 0}$ with a common stationary distribution $\boldsymbol{\pi}$, we may as well consider the product of the sequence of reduced matrices $\left\{\mathbf{A}(t) := \mathbf{W}(t) - \boldsymbol{\pi}\mathbf{1}^{\mathrm{T}}\right\}_{t\geq 0}$ instead. This is due to the following equalities:

$$\prod_{p=1}^{t}\mathbf{W}(t-p) = \prod_{p=1}^{t}\left(\mathbf{A}(t-p) + \boldsymbol{\pi}\mathbf{1}^{\mathrm{T}}\right) = \prod_{p=1}^{t}\mathbf{A}(t-p) + \boldsymbol{\pi}\mathbf{1}^{\mathrm{T}} \tag{3.2.26}$$

$$\delta\left(\prod_{p=1}^{t}\mathbf{W}(t-p)\right) = \delta\left(\prod_{p=1}^{t}\mathbf{A}(t-p)\right). \tag{3.2.27}$$

[1] If $\mathbf{W}$ is not fully diagonalizable, one has to consider a Jordan decomposition instead of the eigenvalue decomposition.

If the parameters $\xi$ and $\gamma$ are defined as in equations (3.2.22) and (3.2.23) respectively, they are metrics for the asymptotic convergence rate of a inhomogeneous Markov chain or equivalently, the asymptotic averaging speed in a time-varying network. With equation (3.2.27), Proposition 3.2.12 and Theorem 3.2.10, these parameters are also equal to

$$\xi = \lim_{t\to\infty} \frac{1}{t} \log \left\| \prod_{p=1}^{t} \mathbf{A}(t-p) \right\| \tag{3.2.28}$$

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \mathrm{E}\left[ \log \left\| \prod_{p=1}^{t} \mathbf{A}(t-p) \right\| \right]. \tag{3.2.29}$$

This representation of the averaging speed as a function of an arbitrary norm of the product of $\mathbf{A}(t)$ will turn out to be much more convenient than using Dobrushin's seminorm $\delta$.

## 3.3 Oseledec's Theorem

Up to now, we only considered scalar metrics to describe the asymptotic "contraction rate" of products of random matrices. The following theorem confirms that this is in fact legitimate, and that the parameter $\xi$ actually determines the convergence speed for *almost every* starting point.

**Theorem 3.3.1.** *(Oseledec's Theorem) Let $\|\bullet\|$ be a norm on $\mathbb{R}^{n\times n}$, and let $\{\mathbf{A}(t)\}_{t\geq 0}$ be a stationary sequence of random matrices satisfying $\mathrm{E}\left[\log^+ \|\mathbf{A}(0)\|\right] < \infty$. $\{\mathbf{A}(\omega,t)\}_{t\geq 0}$ denotes a specific realization, and $\mathbf{M}_t(\omega) := \prod_{p=1}^{t} \mathbf{A}(\omega, t-p)$. Then, the following hold with probability one:*

1) *there is a positive integer $s(\omega)$ and there are real numbers $\lambda^{(1)}(\omega) < \lambda^{(2)}(\omega) < \ldots < \lambda^{(s(\omega))}(\omega)$, where $\lambda^{(1)}$ could be $-\infty$,*
2) *there are linear subspaces $\{0\} = V^{(0)}(\omega) \subset V^{(1)}(\omega) \subset \ldots \subset V^{(s(\omega))}(\omega) = \mathbb{R}^n$ of $\mathbb{R}^n$,*
3) *such that for $1 \leq i \leq s(\omega)$,*

$$\lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t(\omega)\mathbf{x}\| = \lambda^{(i)}(\omega) \qquad \forall \mathbf{x} \in V^{(i)}(\omega) \setminus V^{(i-1)}(\omega). \tag{3.3.1}$$

*Remarks.*

- For a more general, but less transparent statement of the theorem, see the book of Walters [26, Theorem 10.2]. The original proof is by Oseledec [28], who considered the case of invertible matrices. Another proof (not restricted to this case) is given by Raghunathan [30] and Ruelle [29].
- If $\{\mathbf{A}(t)\}_{t\geq 0}$ is ergodic (e.g. if the $\mathbf{A}(t)$ are i.i.d.), $s(\omega) = s$ and $\lambda^{(i)}(\omega) = \lambda^{(i)}$, $i = 1, 2, \ldots s$, are non-random, but the subspaces $V^{(i)}(\omega)$ are still random.
- If $\mathbf{A}(t) = \mathbf{A}$ is a constant (i.e. degenerate random variable), $e^{\lambda^{(i)}}$ are the absolute values of the eigenvalues of $\mathbf{A}$.
- The numbers $\lambda^{(i)}(\omega)$ are called *Lyapunov characteristic exponents* and $V^{(1)}(\omega) \subset V^{(2)}(\omega) \subset \ldots \subset V^{(s(\omega))}(\omega)$ is called the associated *filtration*. The number $m^{(i)} = \dim V^{(i)}(\omega) - \dim V^{(i-1)}(\omega)$ is called the *multiplicity* of $\lambda^{(i)}(\omega)$.
- In relation to Section 3.2, we have $\xi = \lambda^{(s(\omega))}$, and in the ergodic case, $\gamma = \lambda^{(s)}$. See Section 3.4 for details.
- In the introductory statement, "almost every" starting point $\mathbf{x}$ meant all $\mathbf{x} \in \mathbb{R}^n \setminus V^{(s(\omega)-1)}(\omega)$, i.e. all $\mathbf{x}$ outside a proper linear $n - m^{(s(\omega))}$ dimensional subspace of $\mathbb{R}^n$.

In view of this theorem, it is really the largest Lyapunov exponent which will determine the rate at which a system will expand or contract, as long as the starting point is not in some set of volume zero (which may happen with non-zero probability only in degenerate cases). We are mainly interested in contracting systems, i.e. with negative largest Lyapunov exponent. Consequently, from now on we will call this quantity the *contraction rate* of the system. In Chapter 4, we will commit ourselves to find ways to bound this quantity. So far, no easy way has been found to compute it exactly in general. Some results exist only in very special cases, for example for independent matrices whose elements are independent Gaussian, or independent Wiener processes [34].

We will denote the contraction rate of an ergodic system by $\gamma$, to keep a notation consistent with the derivations in this chapter. Finally, let us state explicitly the definition of the contraction rate.

**Definition 3.3.2.** The *contraction rate* of a system governed by an ergodic sequence of matrices $\{\mathbf{A}(t)\}_{t\geq 0}$ is given by

$$\gamma := \lambda^{(s)} = \lim_{t\to\infty} \frac{1}{t} \log \left\| \prod_{p=1}^{t} \mathbf{A}(t-p) \right\| = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\| \,. \tag{3.3.2}$$

## 3.4 A Vector Point of View

**General Case**

So far, the results are formulated in terms of functions of matrices, or more specifically matrix norms. A more illustrative representation is to consider what this means in terms of the action of these matrices on a vector $\mathbf{x}$.

In the general case, with the definition of induced matrix norm

$$\|\mathbf{M}\|_p := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|_p}{\|\mathbf{x}\|_p}, \qquad p = 1, 2, \ldots, \infty \tag{3.4.1}$$

equation (3.3.2) translates as

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \log \left( \sup_{\mathbf{x}(0)\neq 0} \frac{\|\mathbf{M}_t\mathbf{x}(0)\|}{\|\mathbf{x}(0)\|} \right). \tag{3.4.2}$$

**Application to Averaging Problem**

Finally, returning to the averaging problem defined in Chapter 1, we can express the maximal asymptotic relative error in terms of a Lyapunov exponent.

First define and recall some shorthand notations, and note the following equalities:

$$\mathbf{P}_t := \prod_{p=1}^{t} \mathbf{W}(t-p) \tag{3.4.3}$$

$$\mathbf{A}(t) := \mathbf{W}(t) - \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{n} \qquad \Rightarrow \quad \mathbf{A}(t)\mathbf{1} = 0 \text{ and } \mathbf{1}^{\mathrm{T}}\mathbf{A}(t) = 0 \tag{3.4.4}$$

$$\mathbf{M}_t := \prod_{p=1}^{t} \mathbf{A}(t-p) \qquad \Rightarrow \quad \mathbf{M}_t = \mathbf{P}_t - \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{n} \tag{3.4.5}$$

$$\mathbf{y}(t) := \mathbf{x}(t) - \bar{x}\mathbf{1} \qquad \Rightarrow \quad \Big(\mathbf{x}(0) \neq \bar{x}\mathbf{1} \Leftrightarrow \mathbf{y}(0) \neq 0\Big). \tag{3.4.6}$$

With these definitions, the following result is now pretty straightforward. Simply note that $\mathbf{y}(t) = \mathbf{M}_t\mathbf{y}(0)$.

**Theorem 3.4.1.** *In the averaging problem described in Chapter 1 with the notation defined in Section 1.3 and given above, if the sequence $\{\mathbf{W}(t)\}_{t\geq 0}$ is ergodic, the asymptotic maximal relative error*

$$\lim_{t\to\infty}\frac{1}{t}\log\left(\sup_{\mathbf{y}(0)\neq 0}\frac{\|\mathbf{y}(t)\|}{\|\mathbf{y}(0)\|}\right)=\lim_{t\to\infty}\frac{1}{t}\log\left(\sup_{\mathbf{x}(0)\neq\bar{x}\mathbf{1}}\frac{\|\mathbf{x}(t)-\bar{x}\mathbf{1}\|}{\|\mathbf{x}(0)-\bar{x}\mathbf{1}\|}\right) \tag{3.4.7}$$

*is nonrandom and given by*

$$\gamma=\lim_{t\to\infty}\frac{1}{t}\log\left\|\prod_{p=1}^{t}\mathbf{W}(t-p)-\frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{n}\right\|. \tag{3.4.8}$$

To get more insight about this special case, consider Oseledec's Theorem (Theorem 3.3.1), with the ergodic sequence $\{\mathbf{W}(t)\}_{t\geq 0}$ of doubly stochastic matrices $\mathbf{W}(t)$. Then, $\|\mathbf{P}_t\|_1 = \|\mathbf{W}(t)\|_1 = 1$. The largest Lyapunov exponent is then given by

$$\begin{aligned}
\lambda_{\mathbf{W}}^{(s_{\mathbf{W}})} &= \lim_{t\to\infty}\frac{1}{t}\log\|\mathbf{P}_t\mathbf{x}\| && \mathbf{x}\in\mathbb{R}^n\setminus V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)} && (3.4.9)\\
&= \lim_{t\to\infty}\frac{1}{t}\log\frac{\|\mathbf{P}_t\mathbf{x}\|}{\|\mathbf{x}\|} && \mathbf{x}\in\mathbb{R}^n\setminus V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)} && (3.4.10)\\
&= \lim_{t\to\infty}\frac{1}{t}\log\sup_{\mathbf{x}\neq 0}\frac{\|\mathbf{P}_t\mathbf{x}\|}{\|\mathbf{x}\|} && \mathbf{x}\in\mathbb{R}^n\setminus\{0\} && (3.4.11)\\
&= \lim_{t\to\infty}\frac{1}{t}\log\|\mathbf{P}_t\| && && (3.4.12)\\
&= 0. && && (3.4.13)
\end{aligned}$$

If $\mathbf{x}^{\mathrm{T}}\mathbf{1}=0$, $\|\mathbf{P}_t\mathbf{x}\|\to 0$ and therefore we know that $V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}$ is actually the subspace orthogonal to $\mathbf{1}$, i.e. $V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}=\left\{\mathbf{y}\text{ s.t. }\mathbf{y}^{\mathrm{T}}\mathbf{1}=0\right\}$. In this particular case, the first subspace of the filtration is nonrandom. Further, by Theorem 3.3.1, the second Lyapunov exponent is

$$\lambda_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}=\lim_{t\to\infty}\frac{1}{t}\log\|\mathbf{P}_t\mathbf{y}\|,\qquad \mathbf{y}\in V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}\setminus V_{\mathbf{W}}^{(s_{\mathbf{W}}-2)}(\omega) \tag{3.4.14}$$

where $V_{\mathbf{W}}^{(s_{\mathbf{W}}-2)}(\omega)$ is possibly random. But as $\mathbf{y}^{\mathrm{T}}\mathbf{1}=0$, $\mathbf{P}_t\mathbf{y}=\mathbf{M}_t\mathbf{y}$, and the same theorem now applied to the sequence $\{\mathbf{A}(t)\}_{t\geq 0}$ yields

$$\lambda_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}=\lambda_{\mathbf{A}}^{(s_{\mathbf{A}})}=\lim_{t\to\infty}\frac{1}{t}\log\|\mathbf{M}_t\mathbf{y}\|,\qquad \mathbf{y}\in\mathbb{R}^n\setminus V_{\mathbf{A}}^{(s_{\mathbf{A}}-1)}(\omega). \tag{3.4.15}$$

Consequently, we may relate the respective Lyapunov exponents and associated subspaces of the sequences $\{\mathbf{W}(t)\}_{t\geq 0}$ and $\{\mathbf{A}(t)\}_{t\geq 0}$:

$$\begin{aligned}
\lambda_{\mathbf{W}}^{(s_{\mathbf{W}})} &= 0 && (3.4.16)\\
\gamma=\lambda_{\mathbf{W}}^{(s_{\mathbf{W}}-1)} &= \lambda_{\mathbf{A}}^{(s_{\mathbf{A}})}<0 && (3.4.17)\\
V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)} &= \left\{\mathbf{y}\,,\ \mathbf{y}^{\mathrm{T}}\mathbf{1}=0\right\}. && (3.4.18)
\end{aligned}$$

As indicated by equation (3.4.16), the component of $\mathbf{x}$ in the $\mathbf{1}$ direction, orthogonal to $V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}$, is left unchanged. The corresponding Lyapunov exponent $\lambda_{\mathbf{W}}^{(s_{\mathbf{W}})} = 0$ is trivial. Therefore, the measure of interest is the following Lyapunov exponent of $\{\mathbf{W}(t)\}_{t\geq 0}$, which we call the contraction rate and denote by $\gamma$. This is the minimal rate at which any component of $\mathbf{x}$ in $V_{\mathbf{W}}^{(s_{\mathbf{W}}-1)}$ will vanish.

Because it is not very handy to deal with the second largest Lyapunov exponent of a sequence $\{\mathbf{W}(t)\}_{t\geq 0}$ we rather consider the sequence $\{\mathbf{A}(t)\}_{t\geq 0}$ and study its largest Lyapunov exponent $\lambda_{\mathbf{A}}^{(s_{\mathbf{A}})}$, which by definition of this sequence is equal to the conctraction rate $\gamma$.

Finally, it is worth stressing the fact that the supremum in equation (3.4.7) is achieved by almost every starting point $\mathbf{x}(0)$, more precisely by all $\mathbf{x}(0)$ outside of the random set $V_{\mathbf{A}}^{(s_{\mathbf{A}}-1)}(\omega)$, which has zero volume.

**Corollary 3.4.2.** *In Theorem 3.4.1, the supremum is achieved by all* $\mathbf{x}(0)$ *outside of the proper random subspace* $V_{\mathbf{A}}^{(s_{\mathbf{A}}-1)}(\omega) \subset \mathbb{R}^n$.

# 4. Bounds on the Contraction Rate

In this chapter, we will consider sequences $\{\mathbf{A}(t)\}_{t\geq 0}$ of independent, identically distributed random matrices in $\mathbb{R}^{n\times n}$. In the setting of the problem presented in Chapter 1, we can see the matrices as $\mathbf{A}(t) = \mathbf{W}(t) - \mathbf{1}\mathbf{1}^{\mathrm{T}}/n$, but the results we present hold in much more general cases.

From the results of Fuerstenberg and Kesten (Theorem 3.2.8), we know that for a sequence of i.i.d. random matrices, the leading Lyapunov exponent (or contraction rate) $\gamma$ is a degenerate random variable, almost surely equal to its expectation. The problem we address here is that of the computation of this number, which was already mentioned by Kingman in [23, page 897].

More specifically, we want to express, or at least to bound $\gamma$ in terms of some function of the matrix $\mathbf{A}(t)$, its expectation, its moments, or a similar, relatively simple function of it. The following result is a bound on the leading Lyapunov exponent of i.i.d. sequences of random matrices, the inequality being mainly due to the use of Jensen's Theorem at the beginning of the proof.

## 4.1 First Bound on the Contraction Rate

**Theorem 4.1.1.** *(Bound on the Leading Lyapunov Exponent) Let $\{\mathbf{A}(t)\}_{t\geq 0}$ be a sequence of i.i.d. random matrices in $\mathbb{R}^{n\times n}$ satisfying $\mathrm{E}[\log^+ \|\mathbf{A}(0)\|] < \infty$. Then the leading Lyapunov exponent*

$$\gamma := \lim_{t\to\infty} \frac{1}{t} \log \left\| \prod_{p=1}^{t} \mathbf{A}(t-p) \right\|$$

*is bounded by above by*

$$\gamma \leq \frac{1}{2} \log \rho \left( \mathrm{E}[\mathbf{A}(0) \otimes \mathbf{A}(0)] \right).$$

*Remark.* The theorem can be generalized to the case where $\mathbf{A}(t) \in \mathbb{C}^{n\times n}$. We consider the real case only, in order to focus on the main ideas of the proof and not to confuse the reader with technicalities arising in the complex case. However, the reader skilled in the art will easily extend the following proof to the complex case.

*Proof.* First recall Theorem 3.2.8:

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \log \left( \|\mathbf{M}_t\| \right) \quad \text{a.s., and} \tag{4.1.1}$$

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \mathrm{E} \left[ \log \left( \|\mathbf{M}_t\| \right) \right] \tag{4.1.2}$$

where $\mathbf{M}_t$ is defined as previously as $\mathbf{M}_t := \prod_{p=1}^{t} \mathbf{A}(t-p)$. We use Jensen's inequality [35]

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \mathrm{E} \left[ \frac{1}{2} \log \|\mathbf{M}_t\|^2 \right] \tag{4.1.3}$$

$$\leq \lim_{t\to\infty} \frac{1}{2t} \log \mathrm{E} \left[ \|\mathbf{M}_t\|^2 \right]. \tag{4.1.4}$$

As Theorems 3.2.4 and 3.2.8 hold for any choice of norm, we may choose the one most convenient for our purposes, which is the Frobenius norm. Denoting by $m_{ij}$ the $i$-th row, $j$-th column element of $\mathbf{M}_t$,

$$\gamma \leq \lim_{t\to\infty} \frac{1}{2t} \log \mathrm{E}\left[\|\mathbf{M}_t\|_{\mathrm{F}}^2\right] \tag{4.1.5}$$

$$= \lim_{t\to\infty} \frac{1}{2t} \log \mathrm{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}^2\right]. \tag{4.1.6}$$

Express now $\|\mathbf{M}_t\|_{\mathrm{F}}^2$ as a function $f(\bullet)$ of another matrix, namely $\mathbf{M}_t \otimes \mathbf{M}_t$. Formally, $f$ is thus defined as

$$f : \mathbb{R}^{n^2\times n^2} \to \mathbb{R} : \mathbf{B} \mapsto \sum_{i=1}^{n}\sum_{j=1}^{n} b_{i+n(i-1),j+n(j-1)} \tag{4.1.7}$$

where $b_{k,l}$ is the $k$-th row, $l$-th column element of $\mathbf{B}$. In other words, $f(\mathbf{B})$ sums $n^2$ elements of the $n^2 \times n^2$ matrix $\mathbf{B} = \mathbf{M}_t \otimes \mathbf{M}_t$, which are chosen as the $(i,j)$-th element of the $(i,j)$-th block of $\mathbf{B}$, $\mathbf{B}$ being divided in $n^2$ blocks of size $n \times n$, as shown below.

$$\mathbf{B} = \left[\begin{array}{ccc:ccc:c:ccc}
b_{1,1} & \cdots & * & b_{1,n+2} & \cdots & \cdots & \cdots & b_{1,n^2} \\
\vdots & \ddots & & \vdots & & & & \vdots \\
\hdashline
* & & * & * & & & & \\
b_{n+2,1} & \cdots & * & b_{n+2,n+2} & & & & \vdots \\
\vdots & & & & \ddots & & & \\
\hdashline
\vdots & & & & & \ddots & & \vdots \\
\hdashline
\vdots & & & & & & & \\
b_{n^2,1} & \cdots & & \cdots & & \cdots & \cdots & b_{n^2,n^2}
\end{array}\right].$$

Besides the equality

$$f(\mathbf{M} \otimes \mathbf{M}) = \|\mathbf{M}\|_{\mathrm{F}}^2, \tag{4.1.8}$$

this function has the desirable property of being linear:

$$f(\alpha\mathbf{A} + \beta\mathbf{B}) = \alpha f(\mathbf{A}) + \beta f(\mathbf{B}). \tag{4.1.9}$$

Going back to equation (4.1.5), replace $\|\mathbf{M}\|_{\mathrm{F}}^2$ according to (4.1.8). Then, using property (4.1.9), invoke linearity of expectation to interchange $f(\bullet)$ and $\mathrm{E}[\bullet]$:

$$\begin{aligned}
\gamma &\leq \lim_{t\to\infty} \frac{1}{2t} \log \mathrm{E}\left[f(\mathbf{M}_t \otimes \mathbf{M}_t)\right] && (4.1.10)\\
&= \lim_{t\to\infty} \frac{1}{2t} \log f\left(\mathrm{E}\left[\mathbf{M}_t \otimes \mathbf{M}_t\right]\right) && (4.1.11)\\
&= \lim_{t\to\infty} \frac{1}{2t} \log f\left(\mathrm{E}\left[\prod_{p=1}^{t} \mathbf{A}(t-p) \otimes \prod_{p=1}^{t} \mathbf{A}(t-p)\right]\right) && (4.1.12)\\
&= \lim_{t\to\infty} \frac{1}{2t} \log f\left(\mathrm{E}\left[\prod_{p=1}^{t} \Big(\mathbf{A}(t-p) \otimes \mathbf{A}(t-p)\Big)\right]\right), && (4.1.13)
\end{aligned}$$

where we used the distributive property of the Kronecker product, $(\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D})$ in (4.1.13). Now use the i.i.d. property of the matrices $\mathbf{A}(t)$ to interchange the matrix product and expectation

$$\begin{aligned}
\gamma &\leq \lim_{t\to\infty} \frac{1}{2t} \log f\left(\prod_{p=1}^{t} \mathrm{E}\big[\mathbf{A}(t-p) \otimes \mathbf{A}(t-p)\big]\right) && (4.1.14)\\
&= \lim_{t\to\infty} \frac{1}{2t} \log f\big(\mathrm{E}^t\left[\mathbf{A}(0) \otimes \mathbf{A}(0)\right]\big). && (4.1.15)
\end{aligned}$$

To simplify the notation in the following, we define $\mathbf{B} := \mathrm{E}[\mathbf{A}(0) \otimes \mathbf{A}(0)]$. Using linearity of $f$ once again, we have

$$\begin{aligned}
\gamma &\leq \lim_{t\to\infty} \frac{1}{2t} \log f\left(\mathbf{B}^t\right) && (4.1.16)\\
&= \lim_{t\to\infty} \frac{1}{2t} \log f\left(\left\|\mathbf{B}^t\right\| \frac{\mathbf{B}^t}{\|\mathbf{B}^t\|}\right) && (4.1.17)\\
&= \lim_{t\to\infty} \frac{1}{2t} \log \left\|\mathbf{B}^t\right\| f\left(\frac{\mathbf{B}^t}{\|\mathbf{B}^t\|}\right) && (4.1.18)\\
&= \lim_{t\to\infty} \frac{1}{2t} \left(\log \left\|\mathbf{B}^t\right\| + \log f\left(\frac{\mathbf{B}^t}{\|\mathbf{B}^t\|}\right)\right) && (4.1.19)\\
&= \lim_{t\to\infty} \left(\frac{1}{2} \log \left\|\mathbf{B}^t\right\|^{\frac{1}{t}} + \frac{1}{2t} \log f\left(\frac{\mathbf{B}^t}{\|\mathbf{B}^t\|}\right)\right). && (4.1.20)
\end{aligned}$$

It is a well known fact (Gelfand's formula, see e.g. [31, page 299]) that for any matrix norm, and in particular for any $p$-norm, $p \in \{1, 2, \ldots, \infty\}$,

$$\lim_{t\to\infty} \left\|\mathbf{B}^t\right\|_p^{\frac{1}{t}} = \rho(\mathbf{B}) := \max\{|\lambda| : \lambda \in \lambda(\mathbf{B})\}. \qquad (4.1.21)$$

To bound the last term in equation (4.1.20), we choose $p = 1$ ($p = \infty$ would do as well), which yields

$$f\left(\frac{\mathbf{B}}{\|\mathbf{B}\|_1}\right) = \frac{\sum_{k=1}^{n}\sum_{l=1}^{n} b_{k+n(k-1),l+n(l-1)}}{\max\limits_{1\le j\le n^2}\sum_{i=1}^{n^2}|b_{i,j}|} \tag{4.1.22}$$

$$\le \frac{n^2 \max\limits_{1\le i,j\le n^2} b_{i,j}}{\max\limits_{1\le i,j\le n^2}|b_{i,j}|} \le n^2. \tag{4.1.23}$$

Now using successively equations (4.1.23) and (4.1.21) in (4.1.20), we get

$$\gamma \le \lim_{t\to\infty}\left(\frac{1}{2}\log\left\|\mathbf{B}^t\right\|^{\frac{1}{t}} + \frac{1}{2t}\log n^2\right) \tag{4.1.24}$$

$$= \frac{1}{2}\log\lim_{t\to\infty}\left\|\mathbf{B}^t\right\|^{\frac{1}{t}} + 0 \tag{4.1.25}$$

$$= \frac{1}{2}\log\rho(\mathbf{B}) \tag{4.1.26}$$

which concludes the proof. □

*Remark.* The main point of the proof is to represent a (squared) norm of $\mathbf{M}_t$ as a *linear* function $f$ of another function, say $g_1$, of $\mathbf{M}_t$, which allows to subsequently separate the terms of the matrix product, i.e.

$$\|\mathbf{M}_t\|^2 = f(g_1(\mathbf{M}_t)) = f\left(g_2(\mathbf{A}(t-1))\dots g_2(\mathbf{A}(2))g_2(\mathbf{A}(1))g_2(\mathbf{A}(0))\right), \tag{4.1.27}$$

in order to equate

$$\mathrm{E}\left[\|\mathbf{M}_t\|^2\right] = f\left(\mathrm{E}^t\left[g_2(\mathbf{A}(0))\right]\right). \tag{4.1.28}$$

According to that, instead of the choices made in the proof, one could think of choosing the 2-norm, $f(\mathbf{B}) := \mathrm{tr}(\mathbf{B})$ and $g_1(\mathbf{M}) = g_2(\mathbf{M}) := \mathbf{M}^{\mathrm{T}}\mathbf{M}$. Unfortunately, this choice of functions $f$, $g_1$ and $g_2$ do not allow us to rearrange the terms as needed in order to interchange product and expectation, because

$$\mathrm{tr}\left(\mathbf{M}_t^{\mathrm{T}}\mathbf{M}_t\right) \neq \mathrm{tr}\left(\prod_{p=1}^{t}\mathbf{A}(t-p)^{\mathrm{T}}\mathbf{A}(t-p)\right). \tag{4.1.29}$$

Theorem 4.1.1 can also be stated in terms of the action of the matrix sequence on a vector $\mathbf{y}(0)$.

**Corollary 4.1.2.** *Define* $\mathbf{y}(t) := \prod_{u=1}^{t}\mathbf{A}(t-u)\mathbf{y}(0)$, *where* $\{\mathbf{A}(t)\}_{t\ge0}$ *is an i.i.d. sequence of random matrices, and* $\mathrm{E}\left[\log^+\|\mathbf{A}(0)\|\right] < \infty$. *Then for any* $p \ge 1$,

$$\lim_{t\to\infty}\frac{1}{t}\log\left(\sup_{\mathbf{y}(0)\neq0}\frac{\|\mathbf{y}(t)\|_p}{\|\mathbf{y}(0)\|_p}\right) \le \frac{1}{2}\log\rho\left(\mathrm{E}[\mathbf{A}(0)\otimes\mathbf{A}(0)]\right).$$

*Proof.* By definition of the matrix $p$-norm,

$$\sup_{\mathbf{y}(0)\neq 0} \frac{\|\mathbf{y}(t)\|_p}{\|\mathbf{y}(0)\|_p} = \sup_{\mathbf{y}(0)\neq 0} \frac{\left\|\prod_{u=1}^{t} \mathbf{A}(t-u)\mathbf{y}(0)\right\|_p}{\|\mathbf{y}(0)\|_p} = \left\|\prod_{u=1}^{t} \mathbf{A}(t-u)\right\|_p . \tag{4.1.30}$$

Consequently,

$$\lim_{t\to\infty} \frac{1}{t} \log \left( \sup_{\mathbf{y}(0)\neq 0} \frac{\|\mathbf{y}(t)\|_p}{\|\mathbf{y}(0)\|_p} \right) = \gamma \leq \frac{1}{2} \log \rho \left( \mathrm{E}[\mathbf{A}(0) \otimes \mathbf{A}(0)] \right). \tag{4.1.31}$$

□

## 4.2 Simpler, but Looser Bound

The bound derived above, while being very concise in its formulation, still takes a $n^2 \times n^2$ matrix to compute and evaluate its spectral radius. This motivates the following theorem, which is another bound on $\gamma$. However, simulations show that the bound given below can be significantly looser than the first. Refer to simulation results in Chapter 5 for details.

**Theorem 4.2.1.** *Let $\{\mathbf{A}(t)\}_{t\geq 0}$ be a sequence of i.i.d. random matrices in $\mathbb{R}^{n\times n}$ satisfying $\mathrm{E}[\log^+ \|\mathbf{A}(0)\|] < \infty$, and $\gamma$ its leading Lyapunov exponent (see def. 3.3.2). Then $\gamma$ is bounded from above by*

$$\gamma \leq \frac{1}{2} \log \lambda_1 \left( \mathrm{E}[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)] \right).$$

**Corollary 4.2.2.** *Define $\{\mathbf{y}(t)\}_{t\geq 0}$ as in Corollary 4.1.2. Then, for any $p \in \{1, 2, \ldots, \infty\}$,*

$$\lim_{t\to\infty} \frac{1}{t} \log \left( \sup_{\mathbf{y}(0)\neq 0} \frac{\|\mathbf{y}(t)\|_p}{\|\mathbf{y}(0)\|_p} \right) \leq \frac{1}{2} \log \lambda_1 \left( \mathrm{E}[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)] \right).$$

*Proof.* (of Corollary) The Corollary follows readily from the theorem, by the same arguments as in the proof of Corollary 4.1.2. □

*Proof.* (of Theorem) For any nonrandom choice of $\mathbf{y}(0) \in \mathbb{R}^n \setminus \{0\}$, with the first expectation on $\mathbf{y}(t-1)$ and the second on $\mathbf{A}(t-1)$ in (4.2.1),

$$\begin{aligned}
\mathrm{E}\left[\mathbf{y}^{\mathrm{T}}(t)\mathbf{y}(t)\right] &= \mathrm{E}\left[\mathrm{E}\left[\mathbf{y}^{\mathrm{T}}(t-1)\mathbf{A}^{\mathrm{T}}(t-1)\mathbf{A}(t-1)\mathbf{y}(t-1)|\mathbf{y}(t-1)\right]\right] && (4.2.1)\\
&= \mathrm{E}\left[\mathbf{y}^{\mathrm{T}}(t-1)\,\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(t-1)\mathbf{A}(t-1)\right]\mathbf{y}(t-1)\right] && (4.2.2)\\
&\leq \mathrm{E}\left[\lambda_1\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(t-1)\mathbf{A}(t-1)\right]\right)\mathbf{y}^{\mathrm{T}}(t-1)\mathbf{y}(t-1)\right] && (4.2.3)\\
&= \lambda_1\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)\right]\right)\mathrm{E}\left[\mathbf{y}^{\mathrm{T}}(t-1)\mathbf{y}(t-1)\right] && (4.2.4)\\
&\ \ \vdots\\
&\leq \lambda_1^t\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)\right]\right)\mathbf{y}^{\mathrm{T}}(0)\mathbf{y}(0). && (4.2.5)
\end{aligned}$$

Now rearranging some terms, taking the logarithm and dividing by $2t$ on both sides,

$$\frac{1}{2t} \log \left( \frac{\mathrm{E}\left[\mathbf{y}^{\mathrm{T}}(t)\mathbf{y}(t)\right]}{\mathbf{y}^{\mathrm{T}}(0)\mathbf{y}(0)} \right) \leq \frac{1}{2} \log \lambda_1 \left( \mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)\right] \right). \tag{4.2.6}$$

By Jensen's inequality,

$$\frac{1}{2t}\,\mathrm{E}\left[\log\left(\frac{\mathbf{y}^{\mathrm{T}}(t)\mathbf{y}(t)}{\mathbf{y}^{\mathrm{T}}(0)\mathbf{y}(0)}\right)\right] \leq \frac{1}{2}\log\lambda_1\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)\right]\right). \tag{4.2.7}$$

The above equation also holds in the limit as $t \to \infty$:

$$\lim_{t\to\infty}\frac{1}{t}\,\mathrm{E}\left[\log\frac{\|\mathbf{y}(t)\|_2}{\|\mathbf{y}(0)\|_2}\right] \leq \frac{1}{2}\log\lambda_1\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)\right]\right). \tag{4.2.8}$$

Now set $\mathbf{y}(0) = \hat{\mathbf{y}}(0)$ with

$$\hat{\mathbf{y}}(0) = \arg\sup_{\mathbf{y}(0)\neq 0}\frac{\|\mathbf{y}(t)\|_2}{\|\mathbf{y}(0)\|_2} \tag{4.2.9}$$

$$\Rightarrow \log\frac{\|\hat{\mathbf{y}}(t)\|_2}{\|\hat{\mathbf{y}}(0)\|_2} = \log\left\|\prod_{p=1}^{t}\mathbf{A}(t-p)\right\|_2. \tag{4.2.10}$$

By Theorem 3.2.8, we have then

$$\lim_{t\to\infty}\frac{1}{t}\,\mathrm{E}\left[\log\frac{\|\hat{\mathbf{y}}(t)\|_2}{\|\hat{\mathbf{y}}(0)\|_2}\right] = \gamma, \tag{4.2.11}$$

and consequently, equation (4.2.8) implies

$$\gamma \leq \frac{1}{2}\log\lambda_1\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}(0)\mathbf{A}(0)\right]\right). \qquad \square$$

# 5. Simulation Results

## 5.1 Introductory Remarks

### 5.1.1 Different Metrics for Convergence

Different metrics can be used to characterize the rate of convergence (or contraction) of a system whose dynamics are described by a sequence of matrices. In our case, the system to be described is a network of sensors with randomly failing links between the sensors, where the purpose of the network is to compute in a decentralized way the average of the measurements of all nodes in the network. In order to quantify the contraction rate of the system we may consider one of the following metrics:

1) Dobrushin's ergodicity coefficient, or $\delta$-seminorm, defined as

$$\delta(\mathbf{W}) = \frac{1}{2} \max_{j,k} \sum_{i} |w_{i,j} - w_{i,k}| \,. \tag{5.1.1}$$

2) The 2-norm of the residual matrix $\mathbf{M}_t$, that is the product $\prod \mathbf{W}$ of the weighting matrices $\mathbf{W}(t)$ subtracted by the invariant subspace associated with the vector $\mathbf{1}$, i.e.

$$\|\mathbf{M}_t\|_2 = \left\| \prod_{p=1}^{t} \mathbf{W}(t-p) - \frac{\mathbf{1}\mathbf{1}^{\mathrm{T}}}{n} \right\|_2 .$$

3) Any other matrix norm of the residual matrix $\mathbf{M}_t$, for example the Frobenius norm.
4) A norm of the relative error vector $\mathbf{y}(t)/\left\|\mathbf{y}(0)\right\|$ resulting from an arbitrary choice of initial measurement vector $\mathbf{x}(0) = \mathbf{y}(0) + \bar{x}\mathbf{1}$, for example the 2-norm. This metric will depict the actual typical behavior of the system but, in particular its transient behavior, depends on the actual choice of $\mathbf{x}(0)$, and should therefore be regarded as merely illustrative. However, the asymptotic behavior can be generalized to almost all[1] initial vector choices.

As we have seen earlier (see Chapter 3), all these metrics are equivalent and, in the long run, will decay with the same asymptotic rate $\gamma$. This is illustrated by the following example.

**An Example**

As an illustrative example, Figure 5.1 shows in logarithmic scale the evolution of selected error metrics over algorithm iterations (time instants), using the Metropolis weighting algorithm on a random geometric graph (RGG) with 200 nodes, such as depicted in Figure 5.2, with link failure probability $p = 0.9$.

For the definition of the weighting algorithm refer to Chapter 1. The RGG is defined by nodes uniformly distributed on the unit square, 2 nodes being connected iff the distance between them is less than $r$. In the case considered in Figures 5.1 and 5.2, $r = 0.2$ and the number of nodes $n = 200$. The link failure probability $p$ is set to 0.9. This means a given link is active at time

[1] All initial vectors outside of a proper subspace of $\mathbb{R}^n$; see Theorem 3.3.1.

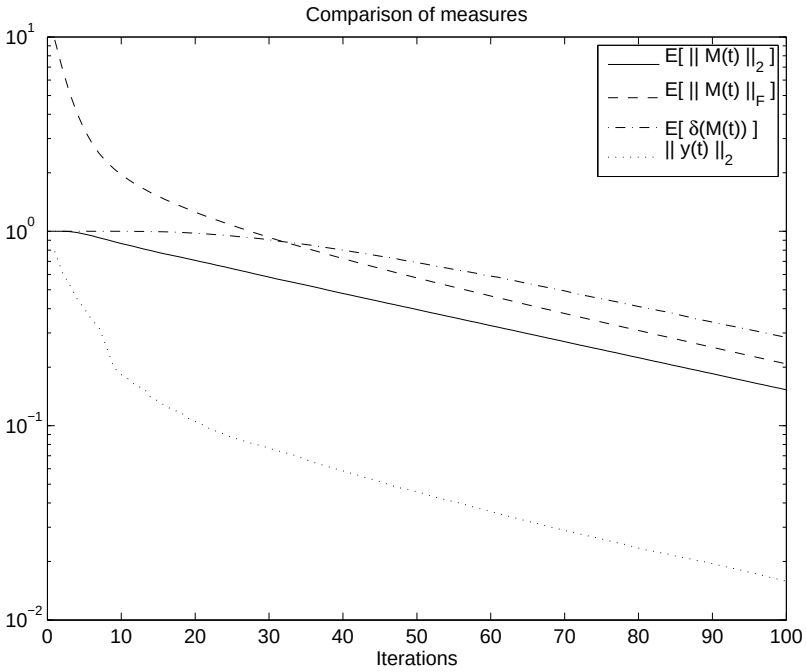


Figure 5.1: Different metrics for the residual error

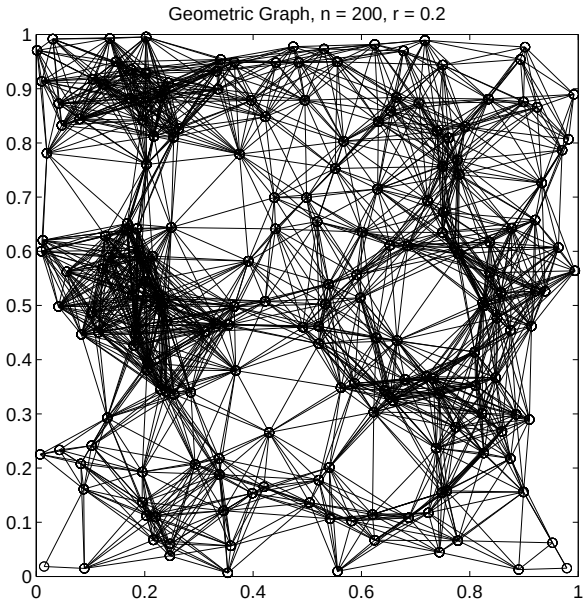


Figure 5.2: Geometric graph with $n = 200$ nodes and $r = 0.2$ maximum link length

instant $t$ with probability $1 - p = 0.1$, independently of other links and independently of its own activity at previous (of future) time instants.

The plot shows the four error metrics described above, namely the Dobrushin-seminorm, the 2-norm and the Frobenius norm of the residual matrix $M_t$, all of them averaged over 400 realizations of link failure/activity sequences. The evolution of the quadratic norm of an error vector $\mathbf{y}(t)$ is also shown, but only one realization for one time sequence, with an initial measurement that was a basis unit vector, i.e. all $n$ nodes had initial value 0, except one node $i$, which was initialized with $x_i = 1$.

Choosing to plot this curve for a specific realization instead of an averaged curve shows the variances involved are very small, and that typical behavior is very close to the mean behavior.

As expected, the slope, i.e. the asymptotic exponential decay rate, of all 4 metrics is the same. Moreover, we can see that the Frobenius-norm approaches the 2-norm. This illustrates Theorem 3.3.1, which states that $\mathbf{M}_t$ tends to become one-dimensional or, in other words, there tends to be only one dominant singular value, and consequently $\|\mathbf{M}_t\|_{\mathrm{F}} = \sum \sigma_i \to \sigma_1 = \|\mathbf{M}_t\|_2$.

Finally, one should note that for an arbitrary initial vector, the error might, in the beginning, decrease much faster than $\|\mathbf{M}_t\|_2$, which by definition always gives the relative error of the "worst case" starting point $\mathbf{y}(0)$.

By a similar argument, one could expect the ergodicity coefficient $\delta$ to have a steeper transient than $\|\mathbf{M}_t\|_2$, because in the case of $\delta$, the maximization goes only over starting vectors of the form $\mathbf{y}(0) = 1/2(\mathbf{e}_i - \mathbf{e}_j)$, $i, j = 1 \dots n$, and not over all $\mathbf{y}(0) \in \mathbb{R}^n$ (see equation (5.1.1)). However, there is another difference between $\delta$ and $\|\mathbf{M}_t\|$. $\delta$ is defined based on the 1-norm,

$$\delta(\mathbf{M}_t) = \max_{i,j} \frac{1}{2} \|\mathbf{M}_t(\mathbf{e}_i - \mathbf{e}_j)\|_1 ,$$

as opposed to $\|\mathbf{M}_t\|_2$ which is defined based on the 2-norm. Because the effect of changing the norm is stronger than that of the constraint on maximization, we observe $\delta(\mathbf{M}_t) > \|\mathbf{M}_t\|_2$.

The main conclusion we draw from this analysis is that, to specify the asymptotic convergence speed of an averaging algorithm, it is sufficient to consider any of the 4 metrics described above. For this reason, in the following we will focus on $\|\mathbf{M}_t\|_2$, and all the results presented for this metric will hold equivalently for the other metrics. The differences observed in the transient are addressed in Section 6.1.2.

### 5.1.2 Scales and Representation

There still are a few precisions to make before going to characterization of the stationary rate of convergence.

The first one is about interchanging expectation and logarithm. Note that in Figure 5.1 we implicitly took the logarithm of the mean of an error metric $\|\mathbf{M}_t\|$ or $\delta(\mathbf{M}_t)$. This is legitimate as long as it is clearly specified that we look at the error norm itself, and not its logarithm.

In all the previously derived results however, we consider the logarithmic error metric $\log \|\mathbf{M}_t\|$ as the value of interest. The difference is very small in practice, so it would not be visible in Figure 5.1 if $\exp\left(\mathrm{E}\left[\log \|\mathbf{M}_t\|\right]\right)$ would have been plotted instead of $\mathrm{E}\left[\|\mathbf{M}_t\|\right]$. But still, it is important to make this distinction. In the following, we will always consider the logarithmic error metric, or in other words, we take logarithms before expectations, i.e. we show quantities such as $\mathrm{E}\left[\|\mathbf{M}_t\|\right]$.

The second point is how to evaluate this stationary contraction rate, i.e. the slope of the curves in Figure 5.1.

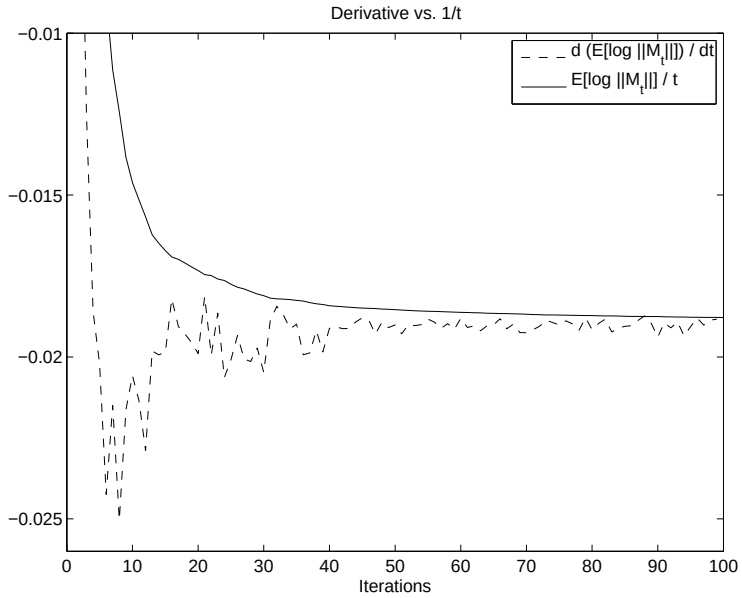


Figure 5.3: $\mathrm{E}\left[\gamma(t)\right]$ and derivative of $\mathrm{E}\left[\log \|\mathbf{M}_t\|_2\right]$, for the same system as before ($n = 200$, Metropolis weights, empirical averaging over 400 realizations).

A first guess might be to use the derivative, e.g. $\mathrm{d}\,\mathrm{E}\left[\log \|\mathbf{M}_t\|\right] / \mathrm{d}\,t$. While this seems intuitively correct, this is not the best thing to do, even if this expectation exists. As shown in Figure 5.3, an empirical averaging over 400 realizations is not sufficient for the derivative (dashed line) to provide an accurate, stable value. For this reason, we will consider the metric

$$\gamma(t) := \frac{1}{t} \log \|\mathbf{M}_t\|_2$$

and its empirical average (solid line in Figure 5.3), which, by theorems in Chapters 3 and 4, is known to be well defined and to converge to a value $\gamma$ as $t \to \infty$, which is such that

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \log \|\mathbf{M}_t\| = \frac{\mathrm{d}\,\mathrm{E}\left[\log \|\mathbf{M}_t\|\right]}{\mathrm{d}\,t},$$

i.e. $\gamma$ characterizes the slope in the stationary region which can be seen in Figure 5.1.

### 5.1.3 A Remark on Rare Events

We consider

$$\gamma = \lim_{t\to\infty} \frac{1}{t} \log \left\| \prod_{p=1}^{t} \mathbf{A}(t-p) \right\|_2 \tag{5.1.2}$$

which by Theorem 3.2.8 is well defined and non-random if $\{\mathbf{A}(t)\}_{t\geq 0}$ is a sequence of i.i.d. matrices.

However, note that if there exists a finite subsequence $\mathbf{M}_T$ of length $T$ of matrices $\mathbf{A}(t)$ such that

$$\mathbf{M}_T = \prod_{p=1}^{T} \mathbf{A}(T-p) = 0 \qquad T < \infty, \tag{5.1.3}$$

which can appear with nonzero probability, it follows immediately that $\gamma = -\infty$. This is because in an infinitely long sequence, the subsequence $\mathbf{M}_T$ will eventually appear, with probability one.

This situation can occur for example when we consider a fully connected graph with independently failing links or nodes. Assume the $e = n(n-1)/2$ links fail independently with probability $p$. Then, there is a probability $(1-p)^e > 0$ that all links are up at the same time. If this happens at time $t$, and we use e.g. the Metropolis weighting algorithm, $\mathbf{W}(t) = \mathbf{1}\mathbf{1}^{\mathrm{T}}/n$ and the exact solution is reached at time $t$, i.e. in finite time.

Another example is when the number of nodes is a power of 2, and the gossip algorithm is used. In this case, the following communication sequence allows all nodes to reach the exact solution in finite time. First, nodes 1 and 2 average their respective values, then nodes 3 and 4, 5 and 6, and so on. In the second stage, we pair up nodes 1 and 3, 2 and 4, 5 and 7, etc... Now, nodes 1 through 4 already have all the same value, and so do nodes 5 – 8 and all subsequent groups of 4. This process can go on, up to the last stage, at which every node from the first half exchanges its value with one of the second half. Consequently, there is a sequence of length $T = n\log(n)/2$ which reaches the exact solution, and therefore $\gamma = -\infty$.

One can find many other examples, in which $\gamma$ is actually not bounded from below. Nevertheless, in many of these cases, simulations will not exhibit this behavior but rather show a finite asymptotic contraction rate. This is because the probability of the event of falling on a sequence like in (5.1.3), while being nonzero, is actually very small, so that with high probability it will not appear in simulations, nor in practice.

It is an instance of the famous "infinite monkey theorem", which states that a monkey typing a random sequence on a typewriter for an infinite amount of time will eventually type out a Shakespeare play (i.e. a specific subsequence of finite length). But like the monkey will probably die before writing even a sonnet, simulations will most probably stop before the occurrence of a sequence satisfying equation (5.1.3).

## 5.2 Analysis of Bounds

After this short introduction motivating our interest for the asymptotic contraction rate, we turn to comparing an empirically obtained rate $\gamma$ to the bounds derived for it in Chapter 4. We will see the first bound is always tighter than the second one, and in many interesting cases much tighter. Also, we will introduce another measure, which is tempting because of its simplicity, and is actually often very close to $\gamma$, but which in general is not a bound, nor even an approximation of $\gamma$.

Recall the two upper bounds on $\gamma$ derived in Chapter 4, Sections 4.1 and 4.2, respectively

$$B_1 := \frac{1}{2}\log\left(\rho\left(\mathrm{E}\left[\mathbf{A}\otimes\mathbf{A}\right]\right)\right) \qquad \text{and} \tag{5.2.1}$$

$$B_2 := \frac{1}{2}\log\left(\rho\left(\mathrm{E}\left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]\right)\right), \tag{5.2.2}$$

and consider also a third measure $C_3$, which was a candidate for approximating $\gamma$:

$$C_3 := \log\left(\rho\left(\mathrm{E}\left[\mathbf{A}\right]\right)\right). \tag{5.2.3}$$

The graphs in the following sections show the evolution of $\gamma(t)$ over time instants at which communication occurs, together with the bounds $B_1$, $B_2$ and the candidate $C_3$.

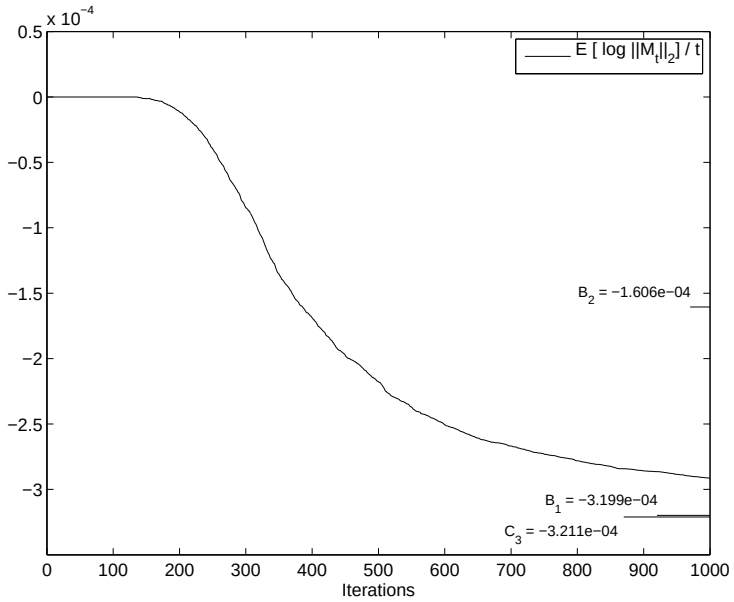


Figure 5.4: $\gamma(t)$, bounds $B_1$, $B_2$ and approximation $C_3$, for a RGG with $n = 60$ and $r = 0.2$, using the gossip algorithm.

### 5.2.1 Geometric Graphs

As an example of a sensor network with randomly placed nodes, consider the case of random geometric graphs (RGG), already presented in the introductory section. More specifically, take a sample RGG, with $n = 60$ nodes uniformly i.i.d. on the unit square, with maximum link length $d = 0.2$.

We considered two different averaging schemes. First, the gossip algorithm, where at each time slot, one link is sampled uniformly from the set of all links and the two incident nodes average their respective values. Figure 5.4. shows the corresponding evolution of $\gamma(t)$ on a time interval of 1000 algorithm iterations, i.e. 1000 communications between nodes, averaged over 500 realizations of link sampling. At $t = 1000$, $\gamma(t)$ has not yet reached its limit $\gamma$. Remember that $B_1 \geq \gamma = \gamma(\infty)$.

In the second case, we consider the "failing links" model, in which at each time instant, links fail (or are inactive) independently of each other and independently of past activity with probability $p = 0.9$ (see example in Section 5.1). A typical configuration of the network at an arbitrary time instant is shown in Figure 5.5, in which currently active links are represented by full lines, and dashed lines denote currently failing (or inactive) links. Note that while the set of full lines is drawn independently at each time instant, the location of the nodes, and therefore the set of all links, remains fixed over time. Notice also that the currently active links typically do not form a connected graph.

In this "failing links" model, we used the Metropolis algorithm (see Chapter 1) to determine the weights on the values exchanged between currently incident nodes. The corresponding results are given in Figure 5.6. Again, $\gamma(t)$ averaged over 500 realizations is plotted against algorithm iterations $t$ up to $t = 1000$, and $B_1$, $B_2$, $C_3$ are indicated on the right. Here, unlike in Figure 5.4, $\gamma(1000)$ is already very close to its limit $\gamma$.

**Observations**

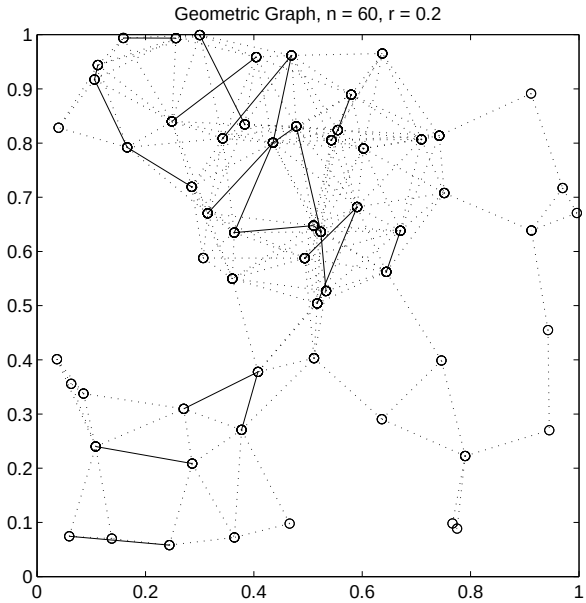


Figure 5.5: Random Geometric Graph (RGG) with $n = 60$ nodes, maximum link length $r = 0.2$, and a proportion of $p = 0.9$ inactive links (dashed).

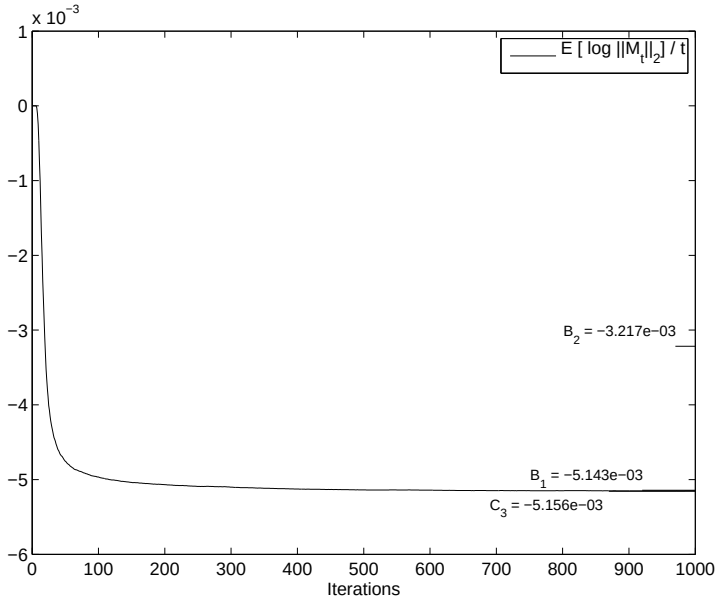


Figure 5.6: $\gamma(t)$, bounds $B_1$, $B_2$ and approximation $C_3$, for a RGG with $n = 60$ and $r = 0.2$, using Metropolis weights, with a proportion of $p = 0.9$ inactive links.

Regarding the evolution of $\gamma(t)$, we can observe a longer transient in the case of the gossip algorithm, as opposed to the failing links model with Metropolis weights. This is not surprising, considering that in the second case many nodes communicate simultaneously, while in the gossip setup there is only one active link at each time step. If one considers the duration of the transient with respect to the total number of communications between nodes, there is almost no noticeable difference. $\gamma(t)$ is less than 0 for the first time after approximately 140 information exchanges in both cases.

Consider now $\gamma$, the value $\gamma(t)$ tends to for large $t$, and compare it to the values $B_1$, $B_2$, and $C_3$. It is more convenient to look at Figure 5.6 rather than Figure 5.4, because it takes longer in the gossip case to reach the limit $\gamma$.

First, one can see that both bounds $B_1$ and $B_2$ are greater than $\gamma$, as expected, but in particular that $B_1$ is significantly tighter than $B_2$. Furthermore, although nothing could be proved concerning the approximation $C_3$, it turns out to be very close to $\gamma$ and $B_1$.

The same observations hold when using the gossip algorithm, as longer simulations have shown. Here one can make an additional observation concerning the looseness of $B_2$. Namely, in the gossip case the weighting matrices are symmetric projection matrices, and therefore $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{A}$. While $C_3$ gives pretty accurately the value of $\gamma$, $B_2$ is exactly $C_3/2$ and consequently a very loose upper bound on $\gamma$.

All these observations were made on RGG's of different sizes, and more generally for all kinds of relatively sparse graphs, e.g. regular graphs with $n = 10$ and degree 3, or $n = 60$ and degree 10. Here, the term "relatively sparse" can be interpreted very widely. Even a RGG with $n = 60$ nodes and maximum link length $r = 0.8$ still exhibited the same behavior and may therefore be considered as "relatively sparse", even though the average node degree is about 30, or $0.5n$.

### 5.2.2 The Full Graph

Consider now a fully connected graph instead of a RGG. This means any two nodes can communicate directly with each other. Consequently, we have a total of $e = n(n-1)/2$ communication links. We consider the same two cases than before. First the gossip algorithm, i.e. there is exactly one active communication link at each time step. Second, the failing links model, where at each time step, every link has a probability $1 - p = 0.1$ of being active.

When using the gossip algorithm in a full graph, the values $\gamma(t)$, $B_1$, $B_2$ and $C_3$ are represented in Figure 5.7, as they were in Figure 5.4 in the RGG case. Similarly, when using the failing links model and the Metropolis weighting algorithm, Figure 5.8 is the equivalent to Figure 5.6 when the RGG is replaced by the full graph. The number of nodes is the same as before, $n = 60$, as well as the number of iterations.

In both cases, gossiping and "failing links", the contraction rate $\gamma$ turns out to be significantly smaller than previously, and also the transient is shorter. This is not surprising, as thanks to "shortcuts" across the graph, differences between distant nodes can be averaged directly, and thus faster than by going through many hops from one neighboring node to the next, as it is necessary in a geometric graph. In the failing links model, due to the increase in the total number of connections, there is also an increase in the number of communications per time step.

More importantly, observe how $B_1$, $B_2$ and $C_3$ have shifted with respect to $\gamma$. Both upper bounds are still greater than or equal to $\gamma$, the limit of $\gamma(t)$ for large $t$, but while $B_2$ was very loose, it is now very close to $B_1$. Conversely, $C_3$ is now clearly less than the empirical convergence rate $\gamma$, and can therefore not be used to approximate it.

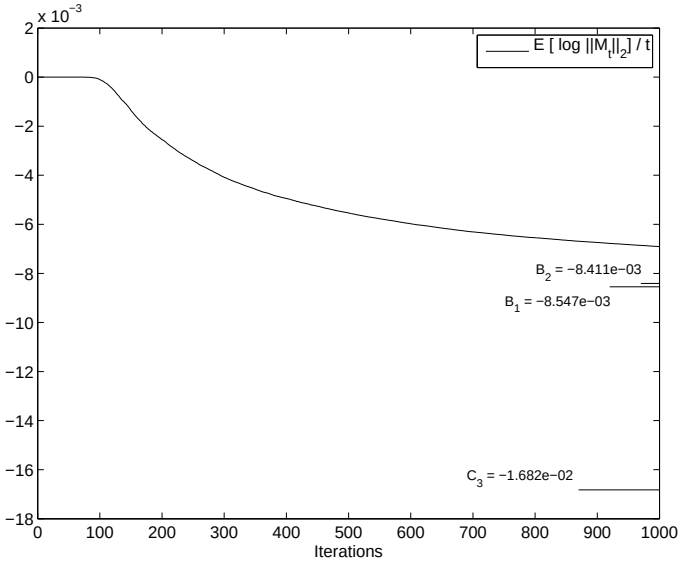


Figure 5.7: $\gamma(t)$, bounds $B_1$, $B_2$ and approximation $C_3$, for a fully connected graph with $n = 60$ nodes, using the gossip algorithm.

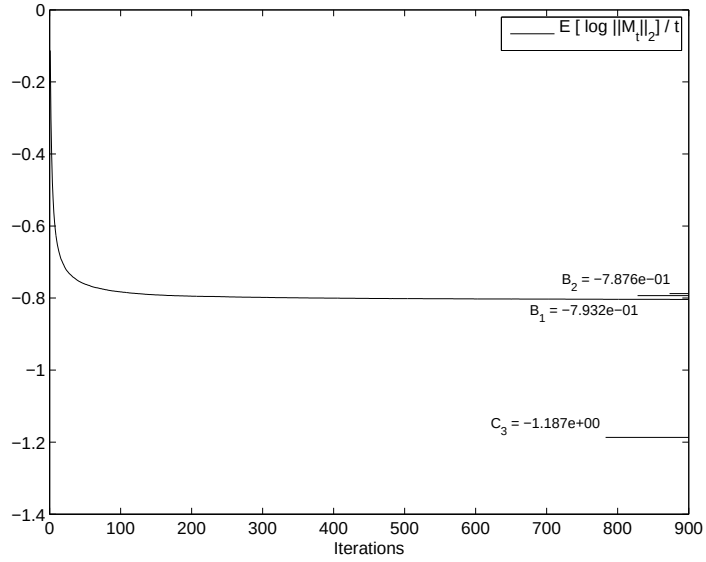


Figure 5.8: $\gamma(t)$, bounds $B_1$, $B_2$ and approximation $C_3$, for a fully connected graph with $n = 60$ nodes, using Metropolis weights, with a proportion of $p = 0.9$ inactive links.

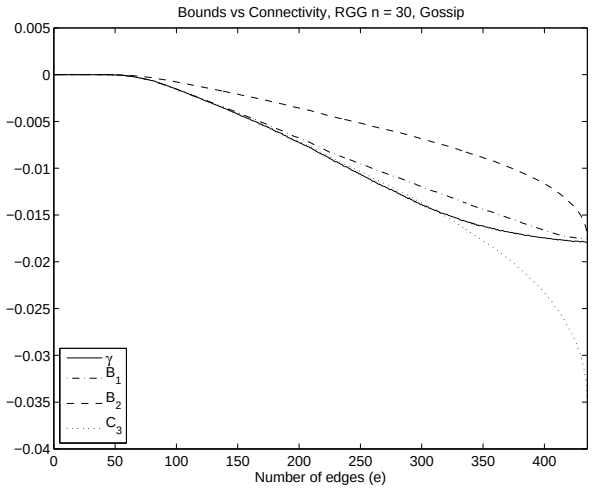


Figure 5.9: Contraction rate $\gamma$, bounds $B_1$, $B_2$ and approximation $C_3$ as a function of the total number of connections, with communications according to the gossip algorithm, for a graph with $n = 30$ nodes.

A legitimate question is whether $C_3$ is a lower bound on $\gamma$. This is not the case. One can easily construct a probability distribution on $\mathbf{A}$ with $\Pr(\mathbf{A} = 0) > 0$, so that $\gamma = -\infty$, but for which $C_3 = \log(\rho(\mathrm{E}[\mathbf{A}]))$ is still finite. Also when both are finite, it can happen that $\gamma < C_3$, as in the case shown in Figure 6.1.

To summarize, $B_2$ is a reasonably tight bound when the graph of possible connections is complete, but rather loose otherwise. Conversely, $C_3$ is a fairly good approximation of $\gamma$ in some cases, where the set of possible connections is limited. In general however, nothing can be said about the relationship between $C_3$ and the asymptotic convergence rate $\gamma$. About $B_1$ we can say that, in all cases studied, it is a reasonably tight bound on the empirical value of $\gamma$. Finally, we want to point out the fact that although in all the simulations, $B_2$ was looser than $B_1$, the question whether this is always the case remains open.

Note also that the actual, mathematically correct value of $\gamma$ is, in many particular cases, not bounded, while the empirical contraction rate computed by simulation is nonetheless finite (see Section 5.1.3). The value considered here is always the empirical contraction rate, not the theoretical one.

### 5.2.3 $\gamma$ as a Function of Graph Density

After having seen how the bounds behave for two extreme configurations, namely barely connected graphs on one side, and the full graph on the other side, it is quite natural to study how these bounds, and $\gamma$, evolve as a function of graph connectivity.

Figure 5.9 shows contraction rate $\gamma$, bounds $B_1$, $B_2$, and approximation $C_3$ as a function of the number of edges, for random geometric graphs with $n = 30$ nodes, when the gossip algorithm is used.

To construct RGG's with a specific number of edges, we proceed as follows. First pick a sample with $n = 30$ nodes uniformly i.i.d. on the unit square. Then add connections between nodes, one after the other, beginning with a connection between the two links which are closest to each other. The next connection to be added is then always between the two nodes which are closest to each other, but not yet directly connected. The bounds are then evaluated when specific numbers of

connections are reached, not necessarily after each added connection. To get Figure 5.9, this whole process was repeated 100 times, with randomly sampled node locations.

When the number of edges $e$ is too small, the graph is not connected, and therefore the sequence of matrices $\{\mathbf{A}(t)\}$ is not contracting, i.e. the Lyapunov exponent $\gamma$ will be equal to 0, as well as both upper bounds. Also the expectation matrix $\mathrm{E}[\mathbf{A}(t)]$ is not contracting as long as the graph is not connected, so $C_3 = 0$ as well.

Then, for increasing number of links and increasing connectivity of the graph, the Lyapunov exponent $\gamma$ will decrease, and so will both upper bounds. In the beginning, $\gamma$ and $B_1$ follows the evolution of $C_3$, but flatten out for very well connected graphs, where $B_2$ and $C_3$ drop very rapidly. Remember that in the gossip case, as represented here, $C_3 = 2B_2$. Consequently, for poorly connected graphs $B_2$ is only about one-half of the actual contraction rate $\gamma$, and therefore very loose. We observe that $B_1$ is always less than, and therefore tighter than $B_2$.

While $B_2$, and also $C_3$, would suggest a big improvement of the convergence speed when filling in the last connections missing in an almost completely connected graph, $B_1$ follows more precisely, although not exactly, the actual contraction rate $\gamma$. The gossip algorithm will not perform significantly better on a full graph than on graphs missing a few links to the complete one.

Of the 3 quantities $B_1$, $B_2$ and $C_3$, the last one is easiest to compute and apparently the most accurate approximation of $\gamma$, except for highly connected graphs. However, this is merely an empirical observation, and at this time there are no means to determine exactly where this approximation holds.

Finally, note that the picture will be very similar in "failing links" scenarios using Metropolis weights (not represented). The qualitative evolution is the same than in Figure 5.9, but the gaps between the curves are much smaller.

## 5.3 Comparison of Averaging Strategies

We will now use the contraction rate $\gamma$ as a metric to compare the performance of different averaging strategies on a given type of network. As we have already seen previously, $\gamma$ is good metric to characterize the averaging performance of a given system, because this is the convergence speed it will have almost surely in the long run, for almost any initial starting vector.

**A Few Definitions**

Let us introduce a couple of parameters related to $\gamma$, which will better illustrate the cost associated with an averaging operation. If the cost factor is time, we may consider the *contraction time* $\mathcal{T}_c$ which we define as

$$\mathcal{T}_c := \frac{-1}{\gamma} = \lim_{t\to\infty} \frac{t}{-\log \|\mathbf{M}_t\|}. \tag{5.3.1}$$

Thus, a low contraction rate will be associated with a large $|\gamma|$ in magnitude, and thus a good, fast averaging performance.

Instead of time, we might be interested in the number of transmitted messages (# of TxM). In systems of autonomous agents connected by wireless links, such as network of remote sensors, the # of TxM is a major cost factor, because every transmission is associated with consumption of energy, which is a scarse resource for remote units. Thus, we define a *contraction energy* as

$$\mathcal{E}_c := \lim_{t\to\infty} \frac{\text{\# of TxM up to time } t}{-\log \|\mathbf{M}_t\|}. \tag{5.3.2}$$

By ergodicity, this is also equal to

$$\mathcal{E}_c = \mathrm{E}\left[\#\text{ of TxM in one iteration}\right] \frac{-1}{\gamma} \tag{5.3.3}$$

$$= \mathrm{E}\left[\#\text{ of TxM in one iteration}\right] \cdot \mathcal{T}_c. \tag{5.3.4}$$

Finally, as we are considering models in which messages can be lost, it might also be interesting to look at the number of received messages (# of RxM). In wireless systems, not only transmitting but also receiving and decoding a message can consume a lot of power. We will see it can make a huge difference to consider received messages instead of transmitted messages. Therefore, we introduce

$$\mathcal{M}_c := \lim_{t\to\infty} \frac{\#\text{ of RxM up to time } t}{-\log \|\mathbf{M}_t\|} \tag{5.3.5}$$

$$= \mathrm{E}\left[\#\text{ of RxM in one iteration}\right] \cdot \mathcal{T}_c. \tag{5.3.6}$$

Again, the second equality follows by ergodicity.

**Comparison of Strategies**

We will compare the costs of the Metropolis weights algorithm with standard pairwise gossip in sparse geometric graphs. More precisely, we consider an ensemble of random geometric graphs (RGGs) on the unit square with $n = 50$ nodes and link range $r = 0.18$. As previously, links between nodes are failing independently in time and independently of each other with probability $p$.

Figures 5.10 through 5.12 show the different performance criteria introduced above as a function of the link failure probability $p$. The represented curves were obtained by averaging the according metric ($\mathcal{E}_c$, $\mathcal{M}_c$ or $\mathcal{T}_c$) over 50 sample graphs in the ensemble specified above.

Before turning to the figures, let us recall the respective averaging algorithms to see how transmitted and received messages are counted. In the standard pairwise gossip algorithm, nodes wake up asynchronously at random times and try to connect to a neighbor in order to exchange their values. Thus, if we consider every instant at which a node wakes up as an iteration, there are successful and unsuccessful iterations. In a successful iteration, two values are exchanged, so we have a total of 2 transmitted and 2 received messages. In the case of an unsuccessful iteration, nothing happens in terms of averaging, but there still is 1 message transmitted (and lost).

The Metropolis algorithm, as opposed to the gossip algorithm, is a synchronized averaging scheme. This means, at each iteration every node in the network starts by sending its current value. Then, every node which received at least one message replies by communicating how many neighbors it has currently, i.e. how many messages it has just received, so that each of its neighbors can choose the appropriate weighting factor for the corresponding value it just received. After these two rounds, every node with at least one active connection to a neighbor knows everything required to perform the averaging, namely its own value, the values of its current neighbors, the size of its own current neighborhood, and the sizes of its current neighbors' neighborhoods. Nodes which did not receive any message in the first round of the iteration may assume they are currently isolated, and do not take any action in the second round. Thus, there are ($n$ + # of not isolated nodes) messages sent at each iteration.

Regarding the number of received messages, we have to take into account that every message sent may be received and decoded zero, one, two, or more times depending on the neighborhood

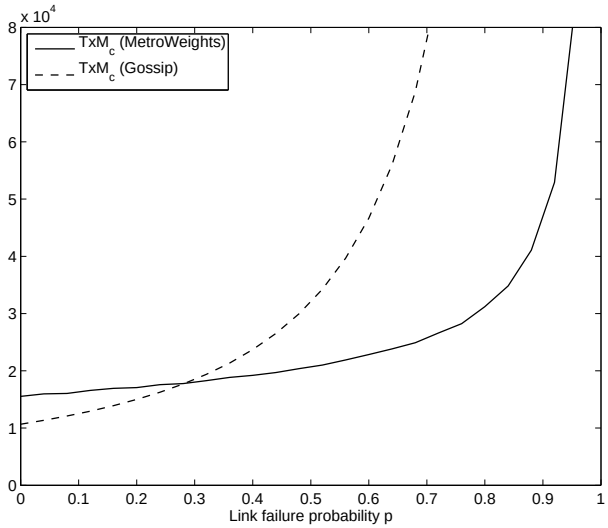


Figure 5.10: Contraction energy $\mathcal{E}_c$, or number of transmitted messages (# of TxM) as a function of link failure probability $p$, for Metropolis weights and gossip algorithms on a RGG.

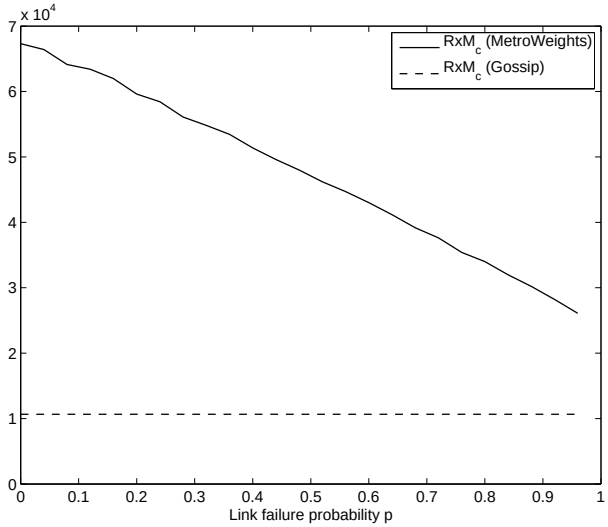


Figure 5.11: $\mathcal{M}_c$, or number of received messages (# of RxM) as a function of link failure probability $p$, for Metropolis weights and gossip algorithms on a RGG.

of the emitting node. Thus, the total number of messages received per round in the whole network is equal to the number of currently active pairwise node to node connections in the network. On average, this will be $p \cdot e$, where $e$ is the total number of node to node links in the network, and $p$ is the failure probability of these links. In the end, as we have two rounds per iteration, one for the current average estimates, and one for the current neighborhood size, the number of received and decoded messages sums up to $2pe$.

Now, one can easily see that counting the transmitted messages (TxM) is radically different from counting the received messages (RxM). Depending on the link failure probability $p$, the number of TxM might be smaller (at low $p$) or greater (at high $p$) than the number of RxM. This can be seen by comparing the associated cost functions respectively shown in Figures 5.10 and 5.11 which both represent the effort associated with a same result.

Figure 5.10 shows the contraction energy $\mathcal{E}_c$ as defined in equation (5.3.2) as a function of the link failure probability, for the two averaging strategies considered, namely the synchronous Metropolis weights algorithm on one hand, and the pairwise gossip algorithm on the other hand. As one could already expect, the required number of messages to transmit to achieve a same result, i.e. a same accuracy of the computed average estimates, grows with the probability $p$, i.e. with the probability of losing emitted messages. A probably less obvious observation is that this number grows faster when using gossiping than with the synchronized strategy.

In Figure 5.11 we see the number of messages received and which need to be decoded corresponding to the number of emitted messages shown in Figure 5.10. As we can see by comparing these two figures, in the gossip algorithm and when no messages are lost, the number of sent and received messages is identical. This does not hold for the Metropolis weighting algorithm, as in this case a single message will typically be received by several nodes. Also, the link failure probability $p$ does not affect the decoding cost of the nodes in gossiping. This comes from the fact that $p$ only affects the number of unsuccessful iterations, and therefore the number of unnecessarily sent messages, but has no effect on successful iterations which are the only which count, both in terms of averaging and decoded messages.

For the Metropolis weighting algorithm, the picture is quite different. The higher $p$ is, i.e. the less connections are active at each round, the more efficient and useful every received message is, and therefore the less messages need to be decoded to achieve the same result. The most striking fact is that this relationship is (or at least seems to be) affine in $p$.

Also, note that in the limit for $p$ close to 1, the metropolis algorithm starts resembling gossiping, at least from the point of view of the averaging operations performed. At high $p$, all links are down with high probability, and only from time to time there will be an active link which will allow two nodes to exchange and average[2] their respective values. As in this case there will be a total of 4 messages received when two nodes communicate, namely both their values and current number of neighbors (=1), as opposed to only 2 messages if we use gossiping, it is not surprising that the limiting point of $\mathcal{M}_c$ (MetroWeights) for $p \to 1$ will be $2\mathcal{M}_c$ (Gossip).

Finally, Figure 5.12 shows contraction time $\mathcal{T}_c$ for the Metropolis weights algorithm, and a $\mathcal{T}_c$ normalized by the number of nodes $n$ for pairwise gossiping. This allows a better comparison between a synchronized algorithm (Metropolis) and an ansynchronous strategy (Gossip). This normalization corresponds to equating the time between two Metropolis iterations and the mean time between wakeups of a node in gossip algorithm.

For all three cost functions considered, we can see that using the gossip algorithm rather than Metropolis weights is advantageous when the link failure probability $p$ is small. For large $p$ however, both time $\mathcal{T}_c$ and energy $\mathcal{E}_c$ required for averaging grow faster when using gossiping. Therefore, it might be advisable to use the Metropolis weighting algorithm when the node to node connections are highly unreliable, and the gossip scheme otherwise.

The cost functions presented and used here are only a few possible examples. Other cost functions might be better suited for particular applications. For example, one might wish to combine $\mathcal{E}_c$ and $\mathcal{M}_c$ in order to take into account the respective costs of both the transmission and the reception/decoding of a message. The resulting cost function would then be a weighted sum of the curves in Figures 5.10 and 5.11. For the Metropolis weights algorithm, this cost function would present a minimum for some $p$ between 0 and 1. Therefore, if one chooses to use this algorithm in

[2]They will actually average them, as both have exactly one neighbor (recall equation (1.2.2)).

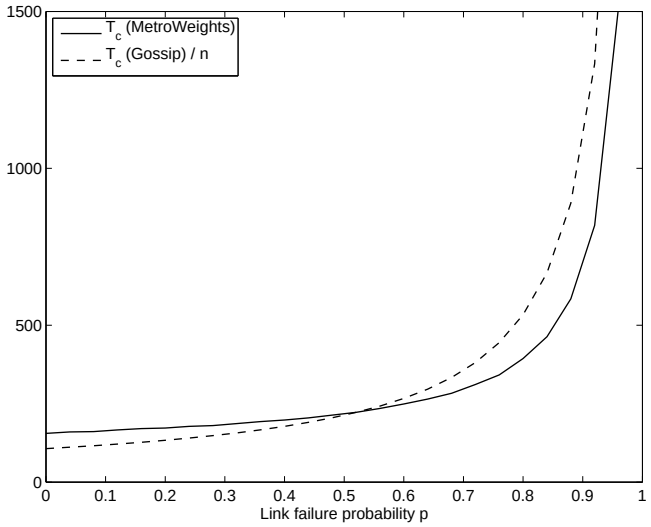


Figure 5.12: Normalized contraction time $\mathcal{T}_c$ as a function of link failure probability $p$, for Metropolis weights and gossip algorithms on a RGG.

a reliable network, it might be advisable to deliberately deactivate a certain number of connections or nodes at each iteration in order to minimize the required power.

In the end, this simple example shows that contraction rate $\gamma$ may be adapted to take into account various cost functions, and can therefore be used to select the best averaging strategy for a given network. It can also help to optimize design parameters of a chosen averaging algorithm, in order to minimize the cost of the averaging operation.

# 6. Summary and Open Problems

## 6.1 Further Observations

During our study on averaging algorithms in time varying networks, we made some interesting observations, which do not fall in the scope of any of the theorems or propositions presented above. Nevertheless, some of these seemed quite interesting to us, and are presented in this section.

Note however that these are only observations based on simulations. In some cases, intuitive arguments which can help understanding the underlying effects are presented, but no formal proofs or theorems could be established.

### 6.1.1 Asymptotically Large Networks

**Bound $B_1$**

The most striking observation about the bounds on the leading Lyapunov exponent $\gamma$ derived in Chapter 4, is the apparent tightness of the first bound

$$B_1 := \frac{1}{2} \log \left( \rho \left( \mathrm{E} \left[ \mathbf{A} \otimes \mathbf{A} \right] \right) \right),$$

as can be seen in Figures 5.4, 5.6, 5.7, 5.8 and 5.9.

However, $B_1$ is not always that close to $\gamma$, as Figure 6.1 shows. This figure shows the evolution of $\gamma(t)$ and the bounds $B_1$, $B_2$, as well as $C_3$, just like the previously cited figures from Chapter 5, but for a much smaller network, with only $n = 10$ nodes, and $e = 13$ edges. One can see the relative gap between $\gamma$ and $B_1$ is much greater than for the larger networks previously considered, which leads us to the following assumption.

**Conjecture.** *$B_1$ is asymptotically tight for large $n$, or in other words, in the limit for $n \to \infty$, $B_1 = \gamma$.*

Although we have no proof for this assertion, here are some thoughts which may help in finding ways to prove it. From Theorem 3.2.8 we know that

$$\gamma := \lim_{t \to \infty} \frac{1}{t} \log \|\mathbf{M}_t\| \tag{6.1.1}$$

$$= \lim_{t \to \infty} \mathrm{E} \left[ \frac{1}{t} \log \|\mathbf{M}_t\| \right] \tag{6.1.2}$$

$$= \lim_{t \to \infty} \mathrm{E} \left[ \log \|\mathbf{M}_t\|^{1/t} \right] \tag{6.1.3}$$

$$= \lim_{t \to \infty} \frac{1}{2} \mathrm{E} \left[ \log \|\mathbf{M}_t\|^{2/t} \right]. \tag{6.1.4}$$

Also, note that since $\gamma$ is a nonrandom real number, so is

$$\begin{aligned} e^{\gamma} &= \exp \left( \lim_{t \to \infty} \log \|\mathbf{M}_t\|^{1/t} \right) \\ &= \lim_{t \to \infty} \exp \left( \log \|\mathbf{M}_t\|^{1/t} \right) = \lim_{t \to \infty} \|\mathbf{M}_t\|^{1/t}. \end{aligned}$$

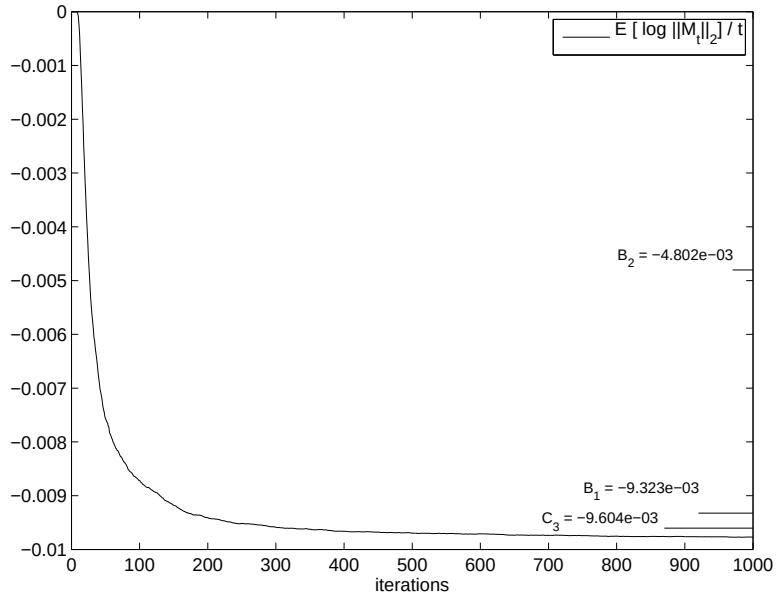


Figure 6.1: $\gamma(t)$, bounds $B_1$, $B_2$ and approximation $C_3$, for a sample graph with $n = 10$ nodes and $e = 13$ links, using the gossip algorithm.

Consequently, $\lim_{n\to\infty} \mathrm{Var}\left(\|\mathbf{M}_t\|^{1/t}\right) = 0$, and in equation (6.1.4), expectation and logarithm can be interchanged while preserving equality:

$$\lim_{t\to\infty} \frac{1}{2} \mathrm{E}\left[\log \|\mathbf{M}_t\|^{2/t}\right] = \lim_{t\to\infty} \frac{1}{2} \log \mathrm{E}\left[\|\mathbf{M}_t\|^{2/t}\right]. \tag{6.1.5}$$

All of the above holds for any sequence $\{\mathbf{M}_t\}_{t\geq 1}$ defined by $\mathbf{M}_t := \prod_{p=1}^{t} \mathbf{A}(t-p)$, where $\{\mathbf{A}(t)\}_{t\geq 0}$ is an ergodic sequence. It holds in particular when $\mathbf{W}(t) = \mathbf{A}(t) + \mathbf{1}\mathbf{1}^{\mathrm{T}}/n$ are i.i.d. weighting matrices of a network with any number of nodes and links.

To us, the crucial point which remains to be cleared seems to be whether

$$\lim_{t\to\infty} \frac{1}{2} \log \mathrm{E}\left[\|\mathbf{M}_t\|^{2/t}\right] = \lim_{t\to\infty} \frac{1}{2} \log \mathrm{E}\left[\|\mathbf{M}_t\|^2\right]^{1/t} \tag{6.1.6}$$

holds, or more precisely to see in which cases it does, and in which cases it does not hold.

**Measure $C_3$**

The same observation of asymptotic tightness can be made about candidate approximation $C_3$, when comparing Figures 5.4 and 6.1. However, the following example will show how tricky the relationship between $\gamma$ and $C_3$ is.

We will consider random geometric graphs (RGGs) with increasing number of nodes $n$, and accordingly decreasing link radius $r$, so as to preserve connectivity when scaling the graph. A result by Gupta and Kumar [36] tells us that for $n \to \infty$, a RGG on a disc[1] of unit area with

$$r^2(n) = \frac{\log(n) + c(n)}{\pi n} \tag{6.1.7}$$

[1]Note the difference to our model, where we use the unit square.

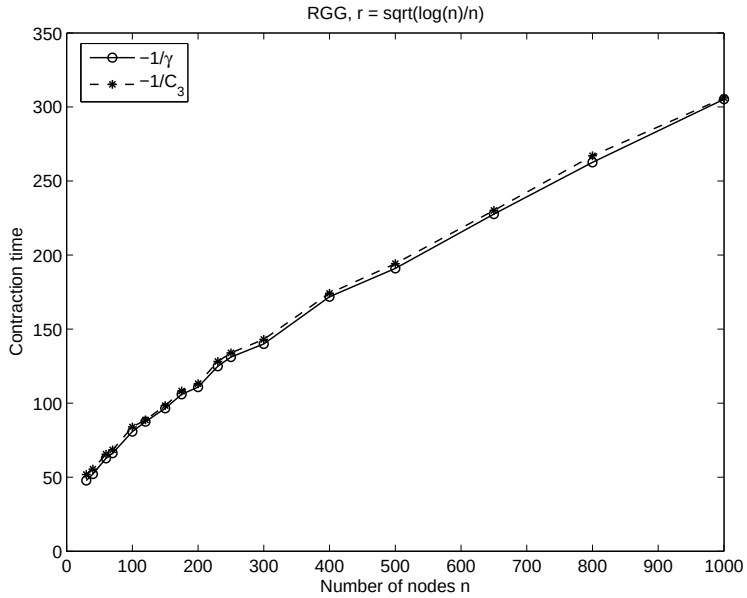


Figure 6.2: Contraction Time $\mathcal{T}_c$ and its "approximation" based on $C_3$, as a function of network size $n$, using Metropolis weights.

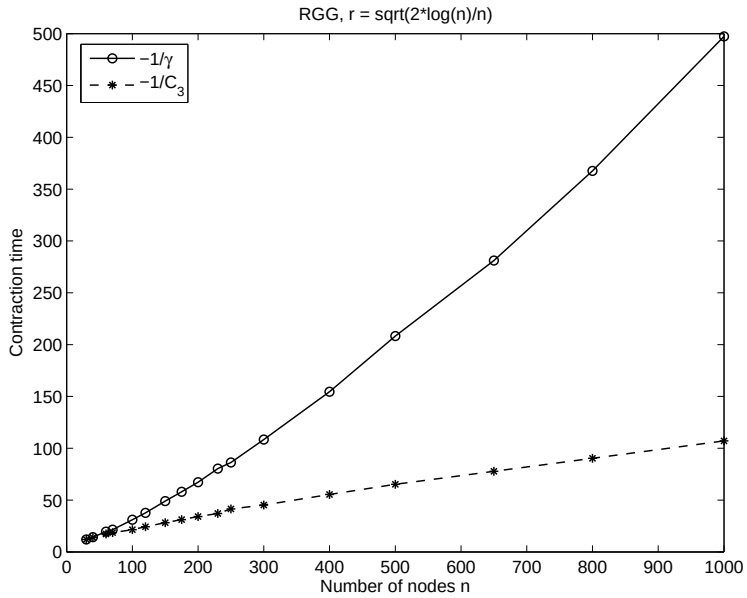


Figure 6.3: Contraction Time $\mathcal{T}_c$ and its "approximation" based on $C_3$, as a function of network size $n$, using Metropolis weights.

is connected iff $c(n) \to \infty$.

We will use the scaling law $r = \sqrt{\alpha \log(n)/n}$ and consider two different values of $\alpha$, namely $\alpha = 1$ and $\alpha = 2$. Note that according to the above result, neither of these values ensures asymptotic connectivity, but nevertheless, at the considered network sizes, this is enough for the networks to be connected with reasonably high probability.

For better readability, we plotted contraction times $\mathcal{T}_c = -1/\gamma$ and its equivalent $-1/C_3$ instead of the contraction rate $\gamma$ itself. Figure 6.2 shows the first case, where $\alpha = 1$, and Figure 6.3 shows the same measures when $\alpha = 2$.

The two pictures are radically different, even though we did not change the order of scaling, but merely a multiplicative coefficient. For $\alpha = 1$ the contraction time $\mathcal{T}_c$ and $-1/C_3$ remain close together, and the according relative error even tends to vanish. On the other hand, if $\alpha = 2$, $C_3$ is no longer a valid approximation for the contraction rate $\gamma$ when $n$ is large.

All this shows that even though intuition tells us there must, or should, be some relationship between $\gamma$ and $C_3$, it is for sure a very complicated one. Even in very similar special cases, $C_3$ and $\gamma$ may in one case be very close, and in the second case behave radically differently. Therefore, at this point we cannot use $C_3$ to approximate the contraction rate.

As a last remark, one could think of an approach to combine the two aspects presented in this section. When (and if!) both $B_1$ and $C_3$ are asymptotically tight, a possible way of proving tightness of one given tightness of the other could be to show both converge to the same value. In order to relate $B_1$ and $C_3$, let us define the autocovariance of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ as the $n^2 \times n^2$ matrix with elements

$$\left(\operatorname{cov}(\mathbf{A})\right)_{(i-1)n+k,(j-1)n+l} = \operatorname{cov}\left(a_{i,j}, a_{k,l}\right). \tag{6.1.8}$$

Then, we can rewrite $B_1$ as

$$B_1 = \frac{1}{2}\log\rho\left(\mathrm{E}\left[\mathbf{A} \otimes \mathbf{A}\right]\right) = \frac{1}{2}\log\rho\left(\mathrm{E}\left[\mathbf{A}\right] \otimes \mathrm{E}\left[\mathbf{A}\right] + \operatorname{cov}(\mathbf{A})\right). \tag{6.1.9}$$

Now, if $\operatorname{cov}(\mathbf{A})$ were 0, we could equate $B_1$ and $C_3$, because

$$\begin{aligned}
\frac{1}{2}\log\rho\left(\mathrm{E}\left[\mathbf{A}\right] \otimes \mathrm{E}\left[\mathbf{A}\right]\right) &= \frac{1}{2}\log\rho^2\left(\mathrm{E}\left[\mathbf{A}\right]\right) && (6.1.10)\\
&= \log\rho\left(\mathrm{E}\left[\mathbf{A}\right]\right) && (6.1.11)\\
&= C_3. && (6.1.12)
\end{aligned}$$

Therefore, $\operatorname{cov}(\mathbf{A})$ might be regarded as a perturbation term. If $\operatorname{cov}(\mathbf{A})$ is somehow "small" compared to $\mathrm{E}\left[\mathbf{A}\right] \otimes \mathrm{E}\left[\mathbf{A}\right]$, chances are $B_1$ and $C_3$ approach each other. However, "small" cannot refer to the magnitude of the elements, as these do not vanish for large $n$. On the other hand, the proportion of nonzero elements in $\operatorname{cov}(\mathbf{A})$ vanishes. But this is not sufficient to prove $\rho\left(\mathrm{E}\left[\mathbf{A} \otimes \mathbf{A}\right]\right) \to \rho\left(\mathrm{E}\left[\mathbf{A}\right] \otimes \mathrm{E}\left[\mathbf{A}\right]\right)$ as $n$ goes to infinity. Thus, it remains to be shown when and whether $B_1$ and $C_3$ converge to the same value.

### 6.1.2 Transient Behavior and Frobenius Norm

Even though this document is dedicated to the study of the leading Lyapunov exponent, which characterizes the asymptotic contraction rate of the system on large time intervals, our simulations also revealed some interesting facts concerning the transient behavior of the system right after

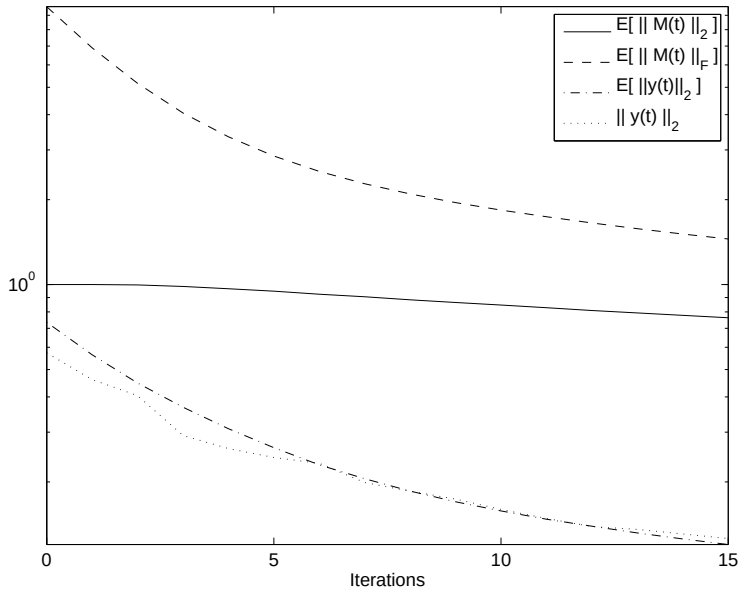


Figure 6.4: Detail of Figure 5.1: Study of the transient

initialization. These observations were already mentioned in Section 5.1.1 of Chapter 5 (Simulation Results), in the discussion of Figure 5.1.

Figure 6.4 is a portion of Figure 5.1 showing in greater detail the beginning of the averaging process. Additionally to the error metrics shown in Figure 5.1, Figure 6.4 includes the average 2-norm of the error vector $\mathbf{y}(t)$, when the initial measurement vector $\mathbf{x}(0)$ is a specific basis unit vector (dash-dotted line). The dotted lines in both figures represent a typical evolution of $\|\mathbf{y}(t)\|$ for the same initial vector, and a specific sequence of active/inactive communication links. (The plot of the Dobrushin-seminorm is omitted in Figure 6.4.) In this figure, one can observe that the average quadratic norm of the error vector $\mathbf{y}(t)$ behaves similarly to the Frobenius norm of the cumulative product matrix $\mathbf{M}(t)$.

To explain this, first recall that the Frobenius norm of a matrix equals the square root of the sum of its squared singular values [31]. Now, if the initial vector $\mathbf{x}(0)$ was a random vector with $n$ Gaussian, zero mean, unit variance, i.i.d. elements ($\sim \mathcal{N}^n(0,1)$), one could use the singular value decomposition $\mathbf{M}_t = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^{\mathrm{T}}$ to see that $\|\mathbf{y}(t)\|_2 = \|\mathbf{M}_t\|_{\mathrm{F}}$ in expectation:

$$\mathbf{y}(t) = \mathbf{M}_t \underbrace{\mathbf{x}(0)}_{\sim\mathcal{N}^n(0,1)} = \overbrace{\mathbf{U}_t}^{\text{unitary}} \underbrace{\mathbf{\Sigma}_t}_{\substack{\text{scales element } i \\ \text{with } \sigma_i}} \underbrace{\overbrace{\mathbf{V}_t^{\mathrm{T}}}^{\text{unitary}}\mathbf{x}(0)}_{\sim\mathcal{N}^n(0,1)}. \tag{6.1.13}$$

One can see that for large $n$, $\|\mathbf{y}(t)\|_2 = \|\mathbf{M}_t\|_{\mathrm{F}}$ holds not only in expectation, but almost surely.

From there, it is intuitively not surprising, that also in the case where the initial vector is a standard basis vector, the error norm decays with a similar rate than the Frobenius norm of the cumulative matrix $\mathbf{M}_t$. Nevertheless, this is not trivial at all. In fact, even though the qualitative behavior is the same, one should still note there are some differences depending on which basis vector is chosen as the starting point. If the node which is initially in disagreement with all the other nodes is badly connected and/or at an extremity of the graph, the convergence will be slower than if that node is well connected. This is only for the transient, of course, because the asymptotic contraction rate will always be the same.

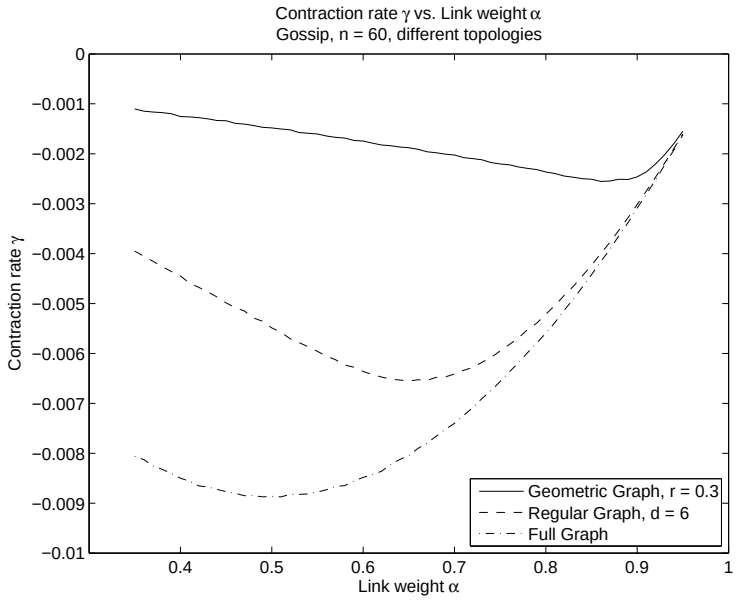


Figure 6.5: $\gamma$ as a function of the message weight in gossip algorithm, for graphs with $n = 60$ nodes, different topologies.

Consequently, the Frobenius norm will, in some cases, give a fairly good idea of the contraction rate, not only asymptotically but also during the transient, for a large range of initial measurement vectors. In the case where the initial measurements are i.i.d. Gaussian, this norm actually precisely represents the expected contraction rate.

### 6.1.3 Gossip – Optimal Weighting

We conclude this series of observations by a last remark concerning gossip algorithms, and more precisely about the choice of the weight on the active link. The usual update strategy in gossip algorithm is that, when two nodes $i$ and $j$ exchange their current values $x_i(t)$ and $x_j(t)$, they both update their estimates of the global average to the local average of the two, i.e.

$$x_i(t+1) = x_j(t+1) = \frac{x_i(t) + x_j(t)}{2}. \tag{6.1.14}$$

While this intuitively seems to be the fastest mixing strategy, this should not be taken for granted in general. In fact, there are many cases in which the above update equation is not the optimal gossip communication (2 by 2 communication) strategy.

Instead of using update equation (6.1.14), consider the following generalization

$$x_i(t+1) = (1-\alpha)x_i(t) + \quad \alpha\ x_j(t) \tag{6.1.15}$$
$$x_j(t+1) = \quad \alpha\ x_i(t) + (1-\alpha)x_j(t), \tag{6.1.16}$$

where $\alpha$ is a positive constant denoted in the following as the *link weight*, and is preferably in the interval $0.5 \leq \alpha < 1$. In this extension, we still consider the link weight to be a constant, the same for all links and for all times, a constant to be specified in advance in the communication protocol. Consequently, this generalization does not incur any additional communication costs when compared to the classical gossip setting, which corresponds to the special case $\alpha = 0.5$.

From an analytical point of view, the only difference is that the update matrices $\mathbf{W}(t)$ are no longer projection matrices. The fact the link weight does not change over time ensures stationarity and ergodicity of the averaging process, which allows us to use the asymptotic contraction rate $\gamma$ to analyse the convergence speed.

Figure 6.5 shows the (approximate) asymptotic contraction rate $\gamma$ as a function of the link weight $\alpha$, for different network topologies. The considered networks are:

1) Full graph: Simplest possible topology model, any node may contact any other node with equal probability.
2) Geometric graph: Nodes are uniformly i.i.d. on the unit square. A node may communicate with any node within a range $r$, in the considered case $r = 0.3$.
3) Regular graph: Every node has exactly the same number of neighbors, but the graph has no geometric structure.

For the plot, we considered networks of size $n = 60$, and computed an approximation of $\gamma$, averaged over 500 different graphs according to the corresponding structure, i.e. either random geometric graph (RGG) or regular graph. Averaging is not necessary for the full graph, as this type of graph has only one realization.

The optimal link weight is in general *not* equal to $0.5$, as can be seen in Figure 6.5. Especially when the graph has a strong geometric structure, i.e. two connected nodes share most of their neighbors, but there are few connections in total, the fastest averaging is achieved at values very close to one, in the considered example around $\alpha = 0.86$. Also in the case where the number of links is limited but there is no geometric structure, the optimal link weight is greater than $0.5$, even though the effect is much weaker than in geometric graphs. In the example given with degree 6, the best contraction rate is achieved around $\alpha = 0.66$. Finally, if the pairs of nodes to communicate are drawn uniformly upon all pairs of nodes, the intuitive choice $\alpha = 0.5$ is the best strategy for averaging.

Inhomogeneous connectivity across the graph explains this phenomenon. In geometric graphs, it is badly connected nodes at the periphery of the graph which limit convergence speed. These nodes get to communicate less often than others, and when they finally do, it might be advantageous for them to basically "take over" the value provided by the better connected neighbor, and to "give away" to that other node their own value, to be diffused to the rest of the network. A similar argument also holds for the regular graph. Although there are no nodes which are better connected than others, there still may exist well connected clusters, which are poorly linked to one another.

Link weight has decidedly a great influence on contraction rate. In many cases, averaging speed can be significantly improved at no cost, by choosing a link weight other than $\alpha = 0.5$. Link weight should be regarded as a design parameter in systems using gossip algorithms. As further work, it might be interesting to investigate how the optimal link weight depends on particular topology parameters, for example average node degree, the girth of the graph, or the maximum connection length in geometric graphs. Further improvements could be achieved at the expense of making the protocol more complicated. Link weight $\alpha$ could for example be modulated as a function of the position of the nodes in the graph. One could also think of changing $\alpha$ as a function of time in order to improve finite time averaging performance.

## 6.2 Summary

The foregoing observations show a few examples how the previously derived tools, in particular the contraction coefficient and its bounds, may be used to study the behavior of averaging algorithms.

These tools were derived in an abstract mathematical framework, which applies to the considered algorithms, but also matches to a variety of other problems. In fact, these tools may be used to study any linear system whose dynamics are governed by a stationary, ergodic sequence of random matrices, regardless of the specific probability distribution of the matrix elements. The conditions for the contraction coefficient $\gamma$ itself to exist are very weak, and basically relate to the boundedness of system parameters, which is given in all physical systems. To apply the bounds derived in Chapter 4, more restrictive assumptions are required. The matrices to be multiplied should not only form an ergodic sequence, but need to be independent and identically distributed. A further, more technical requirement is that the second moments, including cross-moments, of the matrix elements' distributions exist and are finite.

In the considered application, the study of averaging algorithms in time-varying networks, this very general model is restricted to systems governed by sequences of doubly stochastic matrices. However, this very special case actually reverts to the general one, after subtraction of the invariant subspace spanned by the vector $\mathbf{1}$, whose behavior is trivial and of no interest. While product sequences of doubly stochastic matrices are the application we focused on, the model presented in Chapters 3 and 4 may apply to various other random dynamical systems.

Finally, the main advantage of our approach is that it provides a sound, accurate, simple, and theoretically founded method to analyze the performance in terms of convergence speed of a given system. We provide an in depth description of the mathematical tools used, and precise statements of the inferred consequences of these theorems, together with their practical application. The contraction coefficient is a metric for the asymptotic contraction rate of the system, when it runs for a long time. It is simple to compute, as it converges almost surely and not only in expectation, when considering long time intervals. Consequently, a single simulation run is sufficient to determine it for a specific system. We also derived simple, concise upper bounds on the contraction coefficient. Empirical observations have shown one of these is a good bound, and might even be asymptotically tight. Further, the contraction coefficient is a lower bound on the overall contraction rate. A transient phase appearing shortly after initialization of the system and before the stationary regime may lead to a faster convergence rate at the beginning, which rapidly slows down to the asymptotic rate. Finally, our method facilitates a separate study of transient and stationary phenomena.

# Bibliography


[1] Hamid Gharavi and Srikanta P. Kumar. Special issue on sensor networks and applications. *Proceedings of the IEEE*, 91:1151–1153, August 2003.

[2] Ali Jadbabaie, Jie Lin, and A. Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[3] Vincent D. Blondel, Julien M. Hendrickx, Alex Olshevsky, and John N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 2996–3000, December 2005.

[4] Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *International Conference on Information Processing in Sensor Networks*, pages 63–70, Los Angeles, CA, April 2005.

[5] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Gossip algorithms: Design, analysis and applications. In *Proc. IEEE Infocom 2005*, volume 3, pages 1653–1664, Miami, March 2005.

[6] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Trans. Inform. Theory*, 52(6):2508–2530, 2006.

[7] Alexandros G. Dimakis, Anand D. Sarwate, and Martin J. Wainwright. Geographic gossip: efficient aggregation for sensor networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 69–76, New York, NY, USA, 2006. ACM Press.

[8] Mehmet Akar and Robert Shorten. Deterministic synchronization algorithms and convergence rates. In *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 792–796, 2006.

[9] Stephen Boyd, Persi Diaconis, and Lin Xiao. Fastest mixing Markov chain on a graph. *SIAM Rev.*, 46(4):667–689 (electronic), 2004.

[10] Ciamac C. Moallemi and Benjamin Van Roy. Consensus propagation. *IEEE Transactions on Information Theory*, 52(11):4753–4766, November 2006.

[11] Reza Olfati-Saber and Richard M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Automat. Control*, 49(9):1520–1533, 2004.

[12] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems Control Lett.*, 53(1):65–78, 2004.

[13] Pierre Brémaud. *Markov chains*, volume 31 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1999. Gibbs fields, Monte Carlo simulation, and queues.

[14] Lin Xiao, Stephen Boyd, and Seung-Jean Kim. Distributed average consensus with least-mean square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

[15] Lin Xiao and Stephen Boyd. Designing fast distributed iterations via semidefinite programming. Talk presented at Workshop on large scale nonlinear and semidefinite programming, May 2004.

[16] Florence Bénézit. The cost of robustness in average consensus : Slow mixing time, 2006.

[17] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Analysis and optimization of randomized gossip algorithms. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, pages 5310–5315, December 2004.

[18] Lin Xiao, Stephen Boyd, and Sanjay Lall. Distributed average consensus with time-varying metropolis weights. submitted to Automatica, June 2006.

[19] Demosthenis Teneketzis. Lecture notes of EECS 502: Stochastic processes.

[20] Shripad Tuljapurkar. Demography in stochastic environments. ii: growth and convergence rates. *Journal of mathematical biology (J. math. biol.)*, 24(5):569–581, 1986.

[21] Jean-Pierre Eckmann and C.E. Wayne. The largest Liapunov exponent for random matrices and directed polymers in a random environment. *Communications in Mathematical Physics*, 121(1):147–175, March 1989.

[22] J. F. C. Kingman. The ergodic theory of subadditive stochastic processes. *J. Roy. Statist. Soc. Ser. B*, 30:499–510, 1968.

[23] J. F. C. Kingman. Subadditive ergodic theory. *Ann. Probability*, 1:883–909, 1973. With discussion by D. L. Burkholder, Daryl Daley, H. Kesten, P. Ney, Frank Spitzer and J. M. Hammersley, and a reply by the author.

[24] J. F. C. Kingman. Subadditive processes. In *École d'Été de Probabilités de Saint-Flour, V–1975*, pages 167–223. Lecture Notes in Math., Vol. 539. Springer, Berlin, 1976.

[25] Joel E. Cohen. Subadditivity, generalized products of random matrices and operations research. *SIAM Rev.*, 30(1):69–86, 1988.

[26] Peter Walters. *An introduction to ergodic theory*, volume 79 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1982.

[27] H. Furstenberg and H. Kesten. Products of random matrices. *Ann. Math. Statist*, 31:457–469, 1960.

[28] V. I. Oseledec. A multiplicative ergodic theorem. Characteristic Ljapunov, exponents of dynamical systems. *Trudy Moskov. Mat. Obšč.*, 19:179–210, 1968.

[29] David Ruelle. Ergodic theory of differentiable dynamical systems. *Inst. Hautes Études Sci. Publ. Math.*, (50):27–58, 1979.

[30] M. S. Raghunathan. A proof of Oseledec's multiplicative ergodic theorem. *Israel J. Math.*, 32(4):356–362, 1979.

[31] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[32] J. L. Doob. *Stochastic processes*. John Wiley & Sons Inc., New York, 1953.

[33] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[34] Charles M. Newman. The distribution of Lyapunov exponents: exact results for random matrices. *Comm. Math. Phys.*, 103(1):121–126, 1986.

[35] Sheldon Ross. *A first course in probability*. Prentice Hall, Upper Saddle River, fifth edition, 1998.

[36] Piyush Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In G. Yin W. M. McEneany and Q. Zhang, editors, *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, pages 547–566. Birkhauser, Boston, 1998.