Università degli Studi di Pisa

Corso di Laurea Magistrale in Matematica

Tesi di Laurea Magistrale

Numerical methods for computing the steady-state distribution of a G-network.

18 Settembre 2015

Candidato

Tommaso Nesti

Relatori

Prof. **Stefano Giordano**

Prof.ssa. **Beatrice Meini**

Controrelatore

Dott. **Federico Poloni**

Anno Accademico 2014/2015

Contents

Introduction

G-networks are a class of queueing networks introduced by Erol Gelenbe in 1989 ([1]), originally inspired by the spiking behaviour of biophysical neurons. In earlier works (([1],[2],[6])) G-networks were also known as Random Neural Networks, and both the terminologies are still used.

The novelty of G-networks with respect to usual queueing models lies in the presence of *negative customers*, that are used to model requests for *removing work*, in addition to the classical requests for performing work. Basically, a negative customer has the capability to *destroy* a positive customer present in a queue and to disappear instantaneously after that.

A non-trivial feature of G-networks is that, under ergodicity conditions, the steady-state distribution of the number of customers in the network is given as the *product* of the marginal probabilities of the number of customers in each queue. Product form solutions, which are well known to exist for classical queueing networks with only positive customers, such as Jackson networks ([20]) and BCMP networks ([21]), are a desirable trait from a computational and performance evaluation viewpoint.

What distinguishes G-networks from previously known queueing networks is that the equations which yield the arrival rate of customers are nonlinear, making the actual computation of the steady-state distribution a challenging numerical problem. These equations read

$$\begin{cases} \lambda_i^+ = \Lambda_i^+ + \sum_{j=1}^N \mu_j q_j p_{ji}^+ \\ \lambda_i^- = \Lambda_i^- + \sum_{j=1}^N \mu_j q_j p_{ji}^- \qquad i = 1, \dots, N \\ q_i = \min\Big(1, \frac{\lambda_i^+}{\mu_i + \lambda_i^-}\Big). \end{cases} \tag{0.0.1}$$

where $N \in \mathbb{N}$, $\lambda_i^+, \lambda_i^- \in \mathbb{R}^+$ for $i = 1, \dots, N$ are the unknowns and $\mu_j, p_{ij}^+, p_{ij}^-, \Lambda_i^+$ $\Lambda_i^- \in \mathbb{R}^+$ are parameters of the G-network, for $i, j = 1, \dots, N$.

When the system (0.0.1) admits a solution such that $0 < q_i < 1 \quad \forall i = 1, \ldots, N$, the G-network is said to be *stable* and the steady-state distribution is given by $\pi(\mathbf{k}) = \prod_{i=1}^N \pi(k_i) = \prod_{i=1}^N (1-q_i)q_i^{k_i}$ where $\mathbf{k} = [k_1, \ldots, k_N] \in \mathbb{N}^N$ represents the number of customers in the network.

Starting from a real-world application, the *Traffic Matrix Estimation* in large scale IP networks, we observe that developing an efficient method for computing the steady-state distribution would allow to use G-networks with complex topologies, possibly improving the performances.

The main goal of this thesis is to develop efficient numerical methods for computing the solution of equations (0.0.1). We rewrite the system of equations (0.0.1) in matrix form, yielding the equivalent formulation

$$\begin{cases} z = T(z) := \Lambda^+(D_z - P^+)^{-1}P^- D_\mu^{-1} + \alpha D_\mu^{-1} \\ \mathbf{q} := \Lambda^+(D_z - P^+)^{-1} D_\mu^{-1} < \mathbf{1} \end{cases} \tag{0.0.2}$$

where $z \in \mathbb{R}^N$ is the row vectors of unknowns, $\mathbf{1} = [1, \ldots, 1], \Lambda^+, \alpha \in \mathbb{R}^N$ are given row vectors, $P^+, P^-, D_\mu \in \mathbb{R}^{N \times N}$ are nonnegative given matrices and $D_z = \mathrm{diag}(z) \in \mathbb{R}^{N \times N}$ is the diagonal matrix with vector z on the diagonal. Here the symbol $<$ stands for component-wise inequality. Thanks to a result from Gelenbe ([6]), the function $T(z)$ admits a fixed point z^* in the region $\{z \in \mathbb{R}^N : z \geq \mathbf{1}\}$, which may or may not satisfy the ergodicity condition $\mathbf{q} = \Lambda^+(D_{z^*} - P^+)^{-1} D_\mu^{-1} < \mathbf{1}$.

Firstly we propose and analyze the *fixed point iteration* $z^{(k+1)} = T(z^{(k)})$, for $k \geq 0$, starting from a given vector $z^{(0)}$. Under ergodicity condition, we prove that this iteration is locally convergent to the fixed point $z^* \geq \mathbf{1}$, with a linear rate of convergence given by the spectral radius of the Jacobian matrix of the function T at z^*. Moreover, the subsequences $(z^{(2k)})_{k \geq 0}$ and $(z^{(2k+1)})_{k \geq 0}$ satisfy

$$z^{(2k-1)} \leq z^{(2k+1)} \leq z^* \leq z^{2(k+1)} \leq z^{(2k)} \qquad \forall k \geq 0 \tag{0.0.3}$$

i.e. the convergence is alternate around the fixed point, yielding an upper bound for the error of each component at each step.

Secondly, we propose and analyze a *Newton-Raphson* method for the solution of the equation $S(z) := z - T(z) = 0$, namely

$$z^{(k+1)} = z^{(k)} - (z^{(k)} - T(z^{(k)}))(I - J_T(z^{(k)}))^{-1}, \qquad k \geq 0. \tag{0.0.4}$$

We prove that, under stability condition, the iteration (0.0.4) is well defined and locally convergent to the fixed point $z^* \geq \mathbf{1}$ with a quadratic rate, yielding a fast method for the computation of the steady-state distribution.

We will then compare these two methods with an existing algorithm developed by J.M. Fourneau ([4]), concluding that the Newton-Raphson iteration is preferable for moderate values of N. This property makes the Newton-Raphson algorithm an advisable choice for applications where the steady-state distributions of many moderate-sized G-networks have to be computed, as in the Traffic Matrix estimation problem.

The rest of this thesis is structured as follows:

- In Chapter 1 we specify the notation and recall some basic results and definitions.
- In Chapter 2 we describe the G-network model and study its fundamental properties.
- In Chapter 3 we present an application in telecommunication engineering that motivates the need for efficient algorithms.
- Chapter 4 is the core chapter: we recall an existing algorithm and present two new numerical methods for the computation of the stationary distribution.
- Chapter 5 is devoted to numerical experiments, where we compare the performances of the three methods.
- In Chapter 6 we conclude and suggest possible developments.

This thesis was supported by the project PRA 2015 ”Mathematical models and computational methods for complex networks“ of the Department of Computer Science, University of Pisa, coordinated by Prof. Antonio Frangioni.

Chapter 1

Basic tools and definitions

In this chapter we present the basic notation, definitions and results that will be used throughout the thesis.

1.1 Notation

$\mathbb{R}^{m\times n}$	m by n real matrices.
$\mathbb{R}^n_+$	$\{x \in \mathbb{R}^n : x \geq 0\}$.
$A > 0$	for $A \in \mathbb{R}^{m\times n}$, each element of the matrix is greater than zero.
$A \geq 0$	for $A \in \mathbb{R}^{m\times n}$, each element of the matrix is greater than or equal to zero.
$A > B$	$A - B > 0$.
$A \geq B$	$A - B \geq 0$.
I	identity matrix.
$A^\top$	matrix transpose.
A^{-1}	matrix inverse.
$\rho(A)$	spectral radius of matrix A.
$\mathrm{diag}(v)$	diagonal matrix with elements v_i, where $v \in \mathbb{R}^N$.
$B(x,\epsilon)_{\lVert\cdot\rVert}$	for a norm $\lVert\cdot\rVert$ on $\mathbb{R}^N$ and a scalar $\epsilon > 0$, $\{y \in \mathbb{R}^N : \lVert x-y\rVert < \epsilon\}$

1.2 Nonnegative matrices

All matrices considered in this thesis will be real square finite matrices.

Recall that the *spectral radius* of a matrix A is defined as $\rho(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$. Some useful properties of the spectral radius are reported below. Let $N \in \mathbb{N}$.

Lemma 1.2.1. *Let $A \in \mathbb{R}^{N\times N}$. Then, given any $\epsilon > 0$, there exists an induced matrix norm $\|.\|$ such that $\rho(A) \le \|A\| \le \rho(A) + \epsilon$.*

Theorem 1.2.1. *(Gelfand's formula) Let $\|\cdot\|$ be a matrix norm. Then, for all matrices $A \in \mathbb{R}^{N\times N}$, it holds*

$$\rho(A) = \lim_{k\to\infty} \|A^k\|^{1/k}$$

Definition 1.2.1. *A matrix $A \in \mathbb{R}^{N\times N}$ is said to be nonnegative, and we will write $A \ge 0$, if*

$$a_{ij} \ge 0 \quad \forall\, i,j = 1,\ldots,N.$$

Definition 1.2.2. *A matrix $A \in \mathbb{R}^{N\times N}$, with $N \ge 2$, is said to be irreducible if it is not similar via a permutation to an upper block triangular matrix, i.e. there is no real square matrix $P = (p_{ij})$ such that $p_{ij} \in \{0,1\}, P^\top P = I$ and*

$$PAP^\top = \begin{bmatrix} B & 0 \\ C & D \end{bmatrix}$$

where B and D are square matrices of positive size.

A fundamental result regarding nonnegative matrices is the Perron-Frobenius theorem ([18], pp. 26-28).

Theorem 1.2.2 (Perron-Frobenius)**.** *Let $A \in \mathbb{R}^{N\times N}$, $A \ge 0$, be an irreducible matrix. Then the following properties hold:*

1. *$\rho(A) > 0$ and it is an eigenvalue of A;*
2. *There is a vector $v > 0$ such that $Av = \rho(A)v$;*
3. *If $B \ge A$ and $B \ne A$, then $\rho(B) > \rho(A)$;*
4. *$\rho(A)$ is a simple eigenvalue.*

An useful corollary is the following:

Corollary 1.2.1. *Let $A \geq 0$ be an irreducible matrix. Then:*

1. *If the row sums of A are constant, i.e. $\sum_{j=1}^{N} a_{ij} = \sigma \, \forall i = 1, \ldots, N$, then $\rho(A) = \sigma$.*

2. *If the row sums of A have a minimum $\underline{\sigma}$ and a maximum $\overline{\sigma}$, then $\underline{\sigma} < \rho(A) < \overline{\sigma}$.*

Two important classes of matrices which will have an important role in this thesis are the Z-matrices and the M-matrices.

Definition 1.2.3. *$A \in \mathbb{R}^{N \times N}$ is a Z-matrix if $a_{ij} \leq 0 \, \forall i \neq j$.*

Definition 1.2.4. *Let $B \in \mathbb{R}^{N \times N}$,$B \geq 0$ be a nonnegative matrix, and let $s \in \mathbb{R}$. The matrix $A = sI - B$ is an M-matrix if $\rho(B) \leq s$, and it is a non-singular M-matrix if $\rho(B) < s$.*

It is clear that an M-matrix is also a Z-matrix. Regarding the opposite inclusion, the following proposition provides useful criteria for a Z-matrix to be an M-matrix. For the proof and a complete characterization of non-singular M-matrices, see [18].

Theorem 1.2.3 ([18], pp. 1240.)**.** *Let $A \in \mathbb{R}^{N \times N}$ be a Z-matrix. Then A is a non-singular M-matrix if and only if one of the following properties holds:*

1. *The eigenvalues of A have positive real part;*

2. *A is non-singular and A^{-1} is non-negative;*

3. *There exists $x > 0$ such that $Ax > 0$.*

We conclude this section with the following theorem.

Theorem 1.2.4 ([22], pp. 96)**.** *Let $A = M - N$ be a regular splitting of the matrix A, i.e. $M^{-1} \geq 0$ and $N \geq 0$. Then, A is nonsingular with $A^{-1} \geq 0$ if and only if $\rho(M^{-1}N) < 1$, where*

$$\rho(M^{-1}N) = \frac{\rho(A^{-1}N)}{1 + \rho(A^{-1}N)}$$

1.3 Functional Analysis

Definition 1.3.1. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^N$, and $D \subseteq \mathbb{R}^N$ an open subset. A function $F : D \to \mathbb{R}^N$ is said to be Fréchet-differentiable at $x \in D$ if there exists a linear operator $A : \mathbb{R}^N \to \mathbb{R}^N$ such that*

$$\lim_{h \to 0} (1/\|h\|)\|F(x+h) - F(x) - Ah\| = 0$$

In this case, the linear operator A is unique and it is called the Fréchet derivative of F at x. We will write $A = F'$.

Note that the definition of Fréchet-differentiability requires fixing a norm on $\mathbb{R}^N$, but if a function if Fréchet differentiable in a certain norm, then it is Fréchet differentiable in any norm and the Fréchet derivative is the same.

Recall a version of the mean-value theorem, which can be found in [19], pp. 69.

Theorem 1.3.1. *Let $\|\cdot\|$ be a norm on $\mathbb{R}^N$, and let $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ be Fréchet-differentiable on a convex set $D_0 \subset D$ with Fréchet derivative F'. Then, for any $x, y \in D_0$,*

$$\|F(y) - F(x)\| \leq \sup_{0 \leq t \leq 1} \|F'(x + t(y-x))\| \|x - y\|$$

Definition 1.3.2. *A function $F : D \subset \mathbb{R}^N \to \mathbb{R}^N$ is said to be contractive on a subset $D_0 \subseteq D$ if there exists a $0 \leq \alpha < 1$ such that $\|F(x) - F(y)\| \leq \alpha\|x - y\|$ for all $x, y \in D_0$.*

Theorem 1.3.2 (Contraction Mapping Theorem, [19], pp. 120)**.** *Let $F : D \subset \mathbb{R}^N \to \mathbb{R}^N$ be a contractive function on a closed set $D_0 \subset D$, and suppose that $F(D_0) \subset D_0$. Then F has a unique fixed point in D_0.*

Theorem 1.3.3 (Brouwer Theorem, [19], th. 6.3.2.)**.** *Let $F : C \subseteq \mathbb{R}^N \to \mathbb{R}^N$ be continuous on the compact convex set C, and suppose that $F(C) \subseteq C$. Then F has a fixed point in C.*

Chapter 2

The G-network model

2.1 Basic model

The basic G-network model consists of an open network of queues in which two types of customer circulates: positive and negative ones. Each queue consists of one server with i.i.d. exponentially distributed service times, infinite waiting room and First In First Out (FIFO) policy for positive customers.

- Positive customers obey standard service and routing disciplines as in conventional queueing network models. Upon their arrival on a queue, if the server is idle they immediately start being served, otherwise they queue increasing the waiting line length by 1.
- Negative customers behave in the following way: when a negative customer joins a non-empty queue, it destroys one of the present positive customer (in the case of FIFO policy, the destroyed positive customer will be the one who arrived last that queue). If the queue is empty, the negative customer simply vanishes without doing anything else.

  Negative customers are not stored in the queue and they will disappear as soon as they have accomplished their task: as a result, they can not be observed, only the effect of their arrivals can. Finally, negative customers actions are supposed to be taken instantaneously.

Upon completion of service in queue i, the newly served customer either reaches queue j as a positive customer with probability p^+_{ij}, or as a negative customer with probability p^-_{ij}, or it departs from the network with probability d_i. It is important to note that positive customers leaving a queue can become negative when they visit the next queue.

Finally, positive and negative customers can also arrive to queue i from the outside world according to independent Poisson processes with rates Λ_i^+ and Λ_i^- respectively.

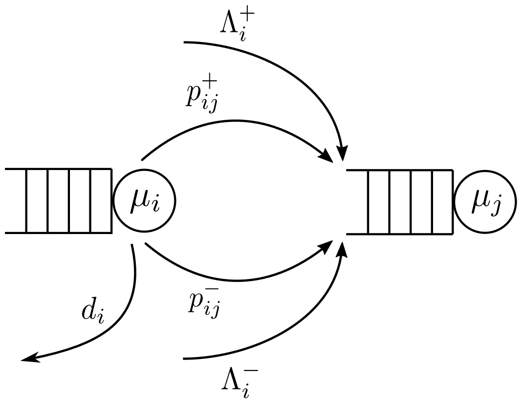

Figure 2.1: The basic G-network model.

2.1.1 Interpretation

From a neural network perspective, positive customers represent *excitation* and negative customers represent *inhibition* of a queue, which is usually called a *neuron* in this setting. The number of positive customers at a neuron, which is a non-negative integer, represents the *potential* of that neuron.

The idea, which motivated the introduction of random neural network in the first place, is that when a neuron is excited, i.e. it has a positive potential, it may "fire", sending signals towards other neurons or outside of the network. As a signal is sent, it reduces the firing neuron's potential and can increase or reduce the receiving neuron's potential.

2.1.2 Extensions and areas of application

The basic G-network model introduced in [1] has been extended in several ways during the years, for example by introducing multiple classes of positive customers, different service policies and different effect of negative customers, such as triggering a customer movement from a queue to another instead of just destroying it ([7],[8],[10]).

G-networks have been applied to several fields, in particular to Combinatorial Optimization ([15]), Image Processing [17] and Telecommunication systems ([14]). The usefulness of G-networks for most of these applications stems from their ability to *learn* from examples and generalize.

In particular, G-networks have been applied ([14]) to the *Traffic Matrix Estimation (TME)* problem in large scale IP networks. We will briefly review this application in Chapter 5 and suggest possible developments.

For a survey on Random Neural Networks and G-networks, see [16].

2.2 Model parameters

In the following, we will always refer to the base G-network model. Let $N \in \mathbb{N}^+$ be the number of queues in a G-network, and recall that when a positive customer leaves queue i it either reaches queue j as a positive customer with probability p_{ij}^+, or as a negative customer with probability p_{ij}^-, or it departs from the network with probability d_i.

These probabilities must sum up to one yielding

$$\sum_{j=1}^{N}(p_{ij}^+ + p_{ij}^-) + d_i = 1 \quad \forall \quad i = 1, \ldots, N \tag{2.2.1}$$

Let $p_{ij} = p_{ij}^+ + p_{ij}^-$ for $i, j = 1, \ldots, N$. The matrix $P = (p_{ij}) \in \mathbb{R}^{N \times N}$ represents the movement of customers between queues.

Customers leaving a queue are not allowed to return directly back to the same neuron, i.e. $p_{ii} = 0$ for all i. Let $P^+ = (p_{ij}^+) \in \mathbb{R}^{N \times N}$ and $P^- = (p_{ij}^-) \in \mathbb{R}^{N \times N}$. The matrices P^+ and P^- are nonnegative, with zero diagonal entries and such that $P = P^+ + P^-$ is row substochastic, i.e.

$$\sum_{j=1}^{N} p_{ij} \leq 1 \quad \forall i = 1, \ldots, N.$$

We assume also that they are irreducible and different from the zero matrix.

We denote by $\Lambda_i^+, \Lambda_i^- \in (0, +\infty)$ be the rates of the Poisson processes representing the arrival of respectively positive and negative customers to queue i. We remind that all this processes are independent of each other. We suppose that at least one of $\Lambda^+ = [\Lambda_1^+, \ldots, \Lambda_N^+]$ and $\Lambda^- = [\Lambda_1^-, \ldots, \Lambda_N^-]$ is different from the zero vector. Finally, let $\mu_i \in (0, +\infty)$ be the rate of the queue i server, meaning that the service time distribution of the single server in queue i has probability density function

$$f(x) = \begin{cases} \mu_i e^{-\mu_i x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

2.3 Steady-state distribution of the number of customers

The state of queue i at time $t \in (0, +\infty)$ is described by the random variable $k_i(t)$, with support $\mathbb{N}$, representing the number of customers present in queue i at time t. These customers are necessarily positive customers, since negative customer, by definition, are never stored in a queue.

The state of the network at time $t \in (0, +\infty)$ is described by the random vector $\mathbf{k}(t) = (k_1(t), \ldots, k_N(t))$, with support $\mathbb{N}^N$.

Letting $\pi(k_i, t) = \mathbb{P}(k_i(t) = k_i)$ and $\pi(\mathbf{k}, t) = \mathbb{P}(\mathbf{k}(t) = \mathbf{k})$, for $k_i \in \mathbb{N}$ and $\mathbf{k} \in \mathbb{N}^N$, we are interested in determining, when they exist, the *steady-state* (or *stationary*) probability distributions for the queues state $\pi(k_i) = \lim_{t\to+\infty} \pi(k_i, t)$ and for the network state $\pi(\mathbf{k}) = \lim_{t\to+\infty} \pi(\mathbf{k}, t)$.

Let λ_j^+ and λ_j^- be the mean arrival rates of respectively positive and negative customers to queue j, in steady-state. They are given by the *traffic equations*

$$\begin{cases} \lambda_i^+ = \Lambda_i^+ + \sum_{j=1}^N \mu_j q_j p_{j,i}^+ \\ \lambda_i^- = \Lambda_i^- + \sum_{j=1}^N \mu_j q_j p_{j,i}^- \\ q_i = \min\left(1, \frac{\lambda_i^+}{\mu_i + \lambda_i^-}\right) \end{cases} \qquad i = 1, \ldots, N \tag{2.3.1}$$

provided that this system admits a solution.

The main result regarding the stationary distribution of a G-network is given in the following theorem, proven by Gelenbe in [1]. It states that if the system of nonlinear equations (2.3.1) admits a solution with $0 < q_i < 1$ for all $i = 1, \ldots, N$, then the stationary distribution of the network state exists and it is given as a product form of the stationary distribution of each queue.

Theorem 2.3.1 ([1]). *Consider a basic G-network with N queues, as described in section 2.1 If the system of non-linear equations*

$$\begin{cases} \lambda_i^+ = \Lambda_i^+ + \sum_{j=1}^N \mu_j q_j p_{j,i}^+ \\ \\ \lambda_i^- = \Lambda_i^- + \sum_{j=1}^N \mu_j q_j p_{j,i}^- \end{cases} \tag{2.3.2}$$

where

$$q_i = \min(1, \frac{\lambda_i^+}{\mu_i + \lambda_i^-}) \tag{2.3.3}$$

admits a unique positive solution $\{\lambda_i^+, \lambda_i^-\}_{i=1}^N$ such that $0 < q_i < 1 \quad \forall\, i = 1, \ldots, N$, then the stationary distributions $\pi(k_i)$ and $\pi(\mathbf{k})$ exist and are given by

$$\pi(k_i) = (1 - q_i) q_i^{k_i} \tag{2.3.4a}$$

$$\pi(\mathbf{k}) = \prod_{i=1}^N \pi(k_i). \tag{2.3.4b}$$

Proof. Let

$$\begin{aligned} \mathbf{k} &= (k_1, \ldots, k_N) \\ \mathbf{k}_i^+ &= (k_1, \ldots, k_{i+1}, \ldots, k_N) \\ \mathbf{k}_i^- &= (k_1, \ldots, k_i - 1, \ldots, k_N) \\ \mathbf{k}_{ij}^{++} &= (k_1, \ldots, k_i + 1, \ldots, k_j + 1, \ldots, k_N) \\ \mathbf{k}_{ij}^{--} &= (k_1, \ldots, k_i - 1, \ldots, k_j - 1, \ldots, k_N) \\ \mathbf{k}_{ij}^{+-} &= (k_1, \ldots, k_i + 1, \ldots, k_j - 1, \ldots, k_N). \end{aligned}$$

The process $\{\mathbf{k}(t)\}_{t \geq 0}$ is a continuous time Markov chain, therefore $\pi(\mathbf{k})$ is described by the steady-state balance equations

$$\begin{aligned} \pi(\mathbf{k}) \sum_{i=1}^N \left[\Lambda_i^+ + (\Lambda_i^- + \mu_i) \mathbf{1}_{\{k_i > 0\}}\right] = \\ \sum_{i=1}^N \Bigg\{ \pi(\mathbf{k}_i^+) \mu_i d_i + \pi(\mathbf{k}_i^-) \Lambda_i^+ \mathbf{1}_{\{k_i > 0\}} + \pi(\mathbf{k}_i^+) \Lambda_i^- + \\ + \sum_{j=1}^N \Big[\pi(\mathbf{k}_{ij}^{+-}) \mu_i p_{ij}^+ \mathbf{1}_{\{k_j > 0\}} + \pi(\mathbf{k}_{ij}^{++}) \mu_i p_{ij}^- + \pi(\mathbf{k}_i^+) \mu_i p_{ij}^- \mathbf{1}_{\{k_j = 0\}} \Big] \Bigg\}. \end{aligned} \tag{2.3.5}$$

This equation is obtained by balancing the exiting and entering flows from state $\mathbf{k}$.

The left hand side represents the rates of exiting flow from state $\mathbf{k}$: any arrival (positive or negative) and every departure causes a transition to another state. The right hand side represents the rates of entering flow to state $\mathbf{k}$. The possible transitions to state $\mathbf{k}$ are:

1. $\mathbf{k}_i^+ \to \mathbf{k}$: it is caused by the arrival of a negative customer from the external to queue i (rate Λ_i^-), the departure of a customer from queue i to the external (rate $\mu_i d_i$) and, in case $k_j = 0$, the arrival of a negative customer from queue i to queue j (rate $\mu_i p_{ij}^- \mathbf{1}_{\{k_j=0\}}$).
2. $\mathbf{k}_i^- \to \mathbf{k}$: it is caused by the arrival of a positive customer from the external (rate $\Lambda_i^+ \mathbf{1}_{\{k_j>0\}}$)
3. $\mathbf{k}_{ij}^{+-} \to \mathbf{k}$: arrival of a positive user departing from queue j in queue i (rate $\mu_i p_{ij}^+ \mathbf{1}_{\{k_j>0\}}$)
4. $\mathbf{k}_{ij}^{++} \to \mathbf{k}$: arrival of a negative user departing from queue j in queue i (rate $\mu_i p_{ij}^-$).

We now verify that the product form equation (2.3.4) satisfies the balance equations. Replacing (2.3.4) in equation (2.3.5) yields

$$\sum_{i=1}^{N} \left[\Lambda_i^+ + (\Lambda_i^- + \mu_i)\mathbf{1}_{\{k_i>0\}}\right] = \sum_{i=1}^{N} \Bigg\{ q_i \mu_i d_i + \frac{\Lambda_i^+}{q_i}\mathbf{1}_{\{k_i>0\}} + \Lambda_i^- q_i + \\ + \sum_{j=1}^{N} \left[\frac{q_i}{q_j}\mu_i p_{ij}^+ \mathbf{1}_{\{k_j>0\}} + q_i q_j \mu_i p_{ij}^- + q_i \mu_i p_{ij}^- \mathbf{1}_{\{k_j=0\}} \right] \Bigg\} \quad (2.3.6)$$

If the solution $\{\lambda_i^+, \lambda_i^-\}$ is such that $0 < q_i < 1 \quad \forall i$, that is $q_i = \frac{\lambda_i^+}{\mu_i + \lambda_i^-} < 1$, we obtain, using (2.3.2)

$$\sum_{i=1}^{N}\left[\Lambda_i^+ + (\Lambda_i^- + \mu_i)\mathbf{1}_{\{k_i>0\}}\right] = \sum_{i=1}^{N}\Big[q_i\mu_i d_i + \frac{\Lambda_i^+}{q_i}\mathbf{1}_{\{k_i>0\}} + \Lambda_i^- q_i\Big] +$$

$$+ \sum_{j=1}^{N}\Big[\frac{\lambda_j^+ - \Lambda_j^+}{q_j}\mathbf{1}_{\{k_j>0\}} + (\lambda_j^- - \Lambda_j^-)q_j + (\lambda_j^- - \Lambda_j^-)\mathbf{1}_{\{k_j=0\}}\Big] =$$

$$= \sum_{i=1}^{N}\Big[q_i\mu_i d_i + \lambda_i^- q_i + \lambda_i^+\mathbf{1}_{\{k_i>0\}} + (\lambda_i^- - \Lambda_i^-)\mathbf{1}_{\{k_i=0\}}\Big] =$$

$$= \sum_{i=1}^{N}\Big[q_i\mu_i - \sum_{j=1}^{N} q_i\mu_i(p_{ij}^+ + p_{ij}^-) + (\mu_i + \lambda_i^-)\mathbf{1}_{\{k_i>0\}} + (\lambda_i^- - \Lambda_i^-)\mathbf{1}_{\{k_i=0\}} + \lambda_i^- q_i\Big].$$

Replacing equation (2.3.2) again yields

$$\sum_{i=1}^{N}\left[\Lambda_i^+ + (\Lambda_i^- + \mu_i)\mathbf{1}_{\{k_i>0\}}\right] =$$

$$\sum_{i=1}^{N}\Big[q_i\mu_i - (\lambda_i^+ - \Lambda_i^+) - (\lambda_i^- - \Lambda_i^-) + (\lambda_i^- - \Lambda_i^-)\mathbf{1}_{\{k_i=0\}} + (\Lambda_i^- + \mu_i)\mathbf{1}_{\{k_i>0\}} + \lambda_i^- q_i\Big] =$$

$$\sum_{i=1}^{N}\Big[q_i\mu_i - (\lambda_i^+ - \Lambda_i^+) + (\Lambda_i^- + \mu_i)\mathbf{1}_{\{k_i>0\}} + \lambda_i^- q_i\Big]$$

which, after using (2.3.3) and canceling terms

$$0 = \sum_{i=1}^{N}\Big[\frac{\lambda_i^+(\mu_i + \lambda_i^-)}{\mu_i + \lambda_i^-} - \lambda_i^+\Big] = 0.$$

Thus, the product form solution is verified since it satisfies the global balance equations (2.3.5). □

This result shows that when a solution to the signal flow equations (2.3.2) can be found such that $0 < q_i < 1$, then the stationary probability distribution of the state of the network exists and can be written as the product of the marginal probabilities of the state of each queue.

Product form solutions are well known to exist for certain networks with only positive customers, such as Jackson networks ([20]) and BCMP network ([21]). What distinguishes this model from previously known queueing networks is that the traffic equations are nonlinear. This property makes the actual computation of the solution of the system (2.3.2) quite challenging, and the main purpose of this thesis is indeed to develop new computational methods for this problem.

Lemma 2.3.1 ([1]). *If a positive solution* $\{\lambda_i^+, \lambda_i^-\}$ *to equations* (2.3.2) *exists with* $0 < q_i < 1 \quad \forall i = 1, \dots, N$*, then it is the unique solution.*

Proof. Since $q_i < 1 \quad \forall i$, the Markov chain $\{\mathbf{k}(t) : t \geq 0\}$ is irreducible and aperiodic. Therefore, if a positive stationary solution $\pi(\mathbf{k})$ exists, then it is unique. By Theorem 2.3.1, $\pi(\mathbf{k})$ exists and is given by (2.3.4), so it is clearly positive for all $\mathbf{k}$. Suppose now there exist two different solutions $\{q_i\}_i$ and $\{q_i'\}_i$ to equations (2.3.2), i.e. $q_j \neq q_j'$ for all $j \in J \subset \{1, \dots, N\}$, $J \neq \emptyset$, and $q_i, q_i' < 1 \; \forall i$. Then $\pi(\mathbf{k}) = \prod_{i=1}^{N}(1 - q_i)q_i^{k_i}$ and $\pi'(\mathbf{k}) = \prod_{i \in J}(1 - q_i')q_i'^{k_i} \prod_{s \in I \setminus J}(1 - q_s')q_s'^{k_s}$ are two different stationary solutions, which is a contradiction. □

Remark 2.3.1. When we talk about the solution of equations (2.3.2) with $q_i < 1$, we can equivalently refer to $\{\lambda_i^+, \lambda_i^-\}_{i=1,\dots,N}$ or to $\{q_i\}_{i=1,\dots,N}$, since the λ_i^+, λ_i^- univocally determine the q_i and vice-versa, thanks to (2.3.2) and (2.3.3)

Remark 2.3.2. In this setting we consider only *open* networks, i.e. networks for which there exist at least a $\Lambda_i^+ > 0$. This means that there is a stream of positive customer from the outside of the networks.

Regarding *closed* network, it can be shown that if P is irreducible and there exists some $p_{uv}^- > 0$, then $\pi(\mathbf{0}) = 1$, that is in steady state the network will be empty almost surely. This is intuitive since there is no external source of positive customers, while there is at least an internal source of negative customers.

As a consequence, the only closed networks which are of interest are those for which either all $p_{ij}^- = 0$ or those for which P is not irreducible, which are not studied in this thesis.

2.4 Network stability

Suppose that a positive solution of equations (2.3.2) exists. Queue i is said to be *stable* if $q_i < 1$. Similarly, the G-network is said to be *stable* if all the queues are stable, that is if $q_i < 1$ for all $i = 1, \ldots, N$.

The reason for such nomenclature is that, as showed in the next corollary of Theorem 2.3.1, whenever a solution of equations (2.3.2) with $0 < q_i < 1$ can be found, all the moments of the network state distribution are finite and can be computed explicitly from the product-form formula in Theorem 2.3.1.

Corollary 2.4.1. *If the G-network is stable, the stationary probability that queue i is busy is given by $\lim_{t\to\infty} \mathbb{P}\{k_i(t) > 0\} = q_i = \frac{\lambda_i^+}{\mu_i+\lambda_i^-}$ and the average number of customer present in queue i in steady-state is $A_i = \mathbb{E}[k_i] = \frac{q_i}{1-q_i}$.*

If for some i we have $q_i = 1$, i.e. $\frac{\lambda_i^+}{\mu_i+\lambda_i^-} \geq 1$, the queue i is called *unstable* or *saturated*. This means that in steady state the queue is constantly excited: $\lim_{t\to+\infty} \mathbb{P}(k_i(t) > 0) = 1$. In this case the stationary distribution does not exist. However, if we restrict ourselves to the sub-network composed only of stable queues, it can be shown that the stationary distribution of this sub-network exists and it is still given in product form. We do not consider this case in this thesis.

2.5 Existence of the stationary distribution

In the previous section we have shown that the stationary distribution of the state of the network exists if there is a solution to (2.3.2) with $0 < q_i < 1$ for all $i = 1, \ldots, N$. Showing the existence of such a solution is a non-trivial task due to the non-linearity of the equations involved.

Partial results about the existence of a solution of equations (2.3.2) were given in [3], while in [6] the existence for the general case was established.

2.5.1 Feed-forward networks

A G-network is said to be *feed-forward* if for any sequence $i_1, \ldots, i_s, \ldots, i_r, \ldots, i_m$ of queues, $i_s = i_r$ for some $r > s$ implies $\prod_{\nu=1}^{m-1} p_{i_\nu i_{\nu+1}} = 0$.

In [3] it is proven that a feed-forward network always admits a unique solution:

Theorem 2.5.1. *If the G-network is feed-forward, then the solution $\{\lambda_i^+, \lambda_i^-\}_{i=1}^N$ of equations* (2.3.2) *exist and it is unique.*

Proof. For any feed-forward network we can renumber the queues so that queue 1 has no predecessors (i.e., $p_{i1} = 0$ for all i), queue N has no successors (i.e. $p_{Ni} = 0$ for all i), and $p_{ij} = 0$ if $j < i$ (recall that $p_{ii} = 0$ for all i in our model). In this way we obtain an network with the property that a customer can go directly form neuron i to neuron j only if $j > i$. This means that the matrices P^+ and P^- are strictly upper triangular matrices, of the form

$$P^+, P^- = \begin{pmatrix} 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.5.1}$$

For such a network, the λ_i^+ and λ_i^- can be computed recursively as follows:

- First set $\lambda_1^+ = \Lambda_1^+, \lambda_1^- = \Lambda_1^-$, and compute $\widetilde{q}_1 = \frac{\lambda_1^+}{\mu_1 + \lambda_1^-}$. If $\widetilde{q}_1 \geq 1$, set $q_1 = 1$ (queue 1 is saturated), otherwise set $q_1 = \widetilde{q}_1$.
- If λ_j^+, λ_j^- for $j = 1, \ldots, i-1$ have been calculated (so also the q_j for $j < i$ have) compute

$$\lambda_i^+ = \Lambda_i^+ + \sum_{j<i} \mu_j q_j p_{j,i}^+$$
$$\lambda_i^- = \Lambda_i^- + \sum_{j<i} \mu_j q_j p_{j,i}^-$$

  and then set $\widetilde{q}_i = \frac{\lambda_i^+}{\mu_i + \lambda_i^-}$. If $\widetilde{q}_i \geq 1$, set $q_i = 1$ (queue i is saturated), otherwise set $q_i = \widetilde{q}_i$.

This process computes in a unique manner the solution of equation (2.3.2), so the theorem is proved. □

In feed-forward network therefore a unique solution always exists, although some queues could be unstable. The computational cost for the algorithm described above is only $O(N^2)$: as a consequence, feed-forward networks have been widely used in applications which requires solving equations (2.3.2) many times, as in supervised learning [6],[14].

2.5.2 Damped networks

In general network stability can not be asserted *a priori*, based on the parameters $\Lambda_i^+, \Lambda_i^-, \mu_i, P^+, P^-$ only, but only after the q_i have been computed.

In some case, however, there are sufficient conditions for network stability which are easy to check.

A G-network is said to be *damped* if the following property holds:

$$\mu_i + \Lambda_i^- > \Lambda_i^+ + \sum_{j=1}^{N} \mu_j p_{ji}^+ \quad \forall i = 1, \ldots, N. \tag{2.5.2}$$

For damped G-networks, we have the following results:

Theorem 2.5.2 ([3]). *If the G-network is damped, the equations* (2.3.2) *always have a unique solution with $q_i < 1$ for all $i = 1 \ldots, N$.*

Condition (2.5.2), although it is quite strong, provides a sufficient condition for network stability which is easy to verify. It can be used to appropriately select parameters of the network to guarantee stability, as in [15].

2.5.3 General case

Solution existence to the general case has been established in [6]. The approach followed is quite general and has also been used to examine solutions existence in extensions of the basic G-network model. To this scope we have to introduce a suitable function $G : \mathbb{R}^N \to \mathbb{R}^N$, as follows.

Rewrite equations (2.3.2) in order to eliminate the q_i terms:

$$\lambda_i^+ = \Lambda_i^+ + \sum_{j=1}^{N} \lambda_j^+ p_{j,i}^+ \frac{\mu_j}{\mu_j + \lambda_j^-}$$
$$\lambda_i^- = \Lambda_i^- + \sum_{j=1}^{N} \lambda_j^+ p_{j,i}^- \frac{\mu_j}{\mu_j + \lambda_j^-}$$

Define the row vectors

$$\lambda^+ = [\lambda_1^+, \ldots, \lambda_N^+]$$
$$\lambda^- = [\lambda_1^-, \ldots, \lambda_N^-]$$
$$\Lambda^+ = [\Lambda_1^+, \ldots, \Lambda_N^+]$$
$$\Lambda^- = [\Lambda_1^-, \ldots, \Lambda_N^-]$$

and let F be the diagonal matrix with elements $f_j = \frac{\mu_j}{\mu_j + \lambda_j^-} > 0$. The previous equations than may be written as

$$\lambda^+(I - FP^+) = \Lambda^+$$
$$\lambda^- = \lambda^+ FP^- + \Lambda^-$$

Matrices P^+ and P^- are irreducible and sub-stochastics, and both of them have at least one row having sum less than 1, because otherwise, due to (2.2.1), one of them would be the zero matrix. Therefore by Corollary 1.2.1 we have $\rho(P^+), \rho(P^-) < 1$. Since $0 \leq P$ and $0 \leq F \leq 1$, we have $0 \leq FP^+ \leq P^+$. According to the Perron-Frobenius theorem 1.2.2, we get $\rho(FP^+) \leq \rho(P^+) < 1$.

Therefore matrix $I - FP^+$ is non-singular and

$$(I - FP^+)^{-1} = \sum_{n=0}^{\infty} (FP^+)^n$$

Consider now the variable $y = \lambda^- - \Lambda^-$, so that

$$F = F(y) = \mathrm{diag}((f_j(y))), \quad f_j(y) = \frac{\mu_j}{\mu_j + \Lambda_j^- + y_j}, \quad y_j = \lambda_j^- - \Lambda_j^- \geq 0.$$

The system (2.3.2) can then be written in the fixed-point form

$$y = G(y)$$

where $G : \mathbb{R}^N \to \mathbb{R}^N$ is given by

$$G(y) = \Lambda^+(I - F(y)P^+)^{-1} F(y) P^-. \tag{2.5.3}$$

Theorem 2.5.3 ([6])**.** *Consider a basic G-network with N queues. Let Λ_i^+, Λ_i^- be the rates of the Poisson processes representing the arrival of positive and negative customers to queue i, μ_i the service rate of queue i and $p_{ji}^+, p_{j,i}^-$ the transition probabilities, as described in section 2.1.1. Then a non-negative solution $\{\lambda_i^+, \lambda_i^-\}_{i=1,\dots,N}$ to equations* (2.3.2)

$$\lambda_i^+ = \Lambda_i^+ + \sum_{j=1}^{N} \mu_j q_j p_{j,i}^+$$
$$\lambda_i^- = \Lambda_i^- + \sum_{j=1}^{N} \mu_j q_j p_{j,i}^-$$

always exists, where $q_i = \frac{\lambda_i^+}{\mu_i + \lambda_i^-}$.

Proof. We will show that the function $G(y)$ defined in (2.5.3) has at least a fixed point $y^* \geq 0$, yielding a solution of equations (2.3.2).

Since $0 \leq F(y) \leq I$ and $0 \leq (F(y)P^+)^n \leq (P^+)^n$ for all $n \in \mathbb{N}$, we have

$$(I - F(y)P^+)^{-1} = \sum_{n=0}^{\infty}(F(y)P^+)^n \leq \sum_{n=0}^{\infty}(P^+)^n = (I - P^+)^{-1}$$

and also

$$0 \leq G(y) = \Lambda^+(I - F(y)P^+)^{-1}F(y)P^- \leq \Lambda^+(I - P^+)^{-1}P^- = G(0) \quad (2.5.4)$$

yielding that G is bounded, i.e. $\|G(y)\| \leq \delta$ for all $y \geq 0$.

G is also Lipschitz continuous: for all $y, z \in \mathbb{R}^+$,

$$\begin{aligned} G(y) - G(z) = \Lambda^+\Big[(F(y)^{-1} - P^+)^{-1} - (F(z)^{-1} - P^+)^{-1}\Big]P^- = \\ \Lambda^+(F(z)^{-1} - P^+)^{-1}\Big[F(z)^{-1} - F(y)^{-1}\Big](F(z)^{-1} - P^+)^{-1} = \\ \Lambda^+(F(z)^{-1} - P^+)^{-1}\Big[\mathrm{diag}\Big(\frac{z_j - y_j}{\mu_j}\Big)\Big](F(z)^{-1} - P^+)^{-1} \end{aligned} \quad (2.5.5)$$

which yields

$$\begin{aligned} &\|G(y) - G(z)\|_\infty \leq \\ &\|(F(z)^{-1} - P^+)^{-1}\|_\infty^2 \|\mathrm{diag}\Big(\frac{z_j - y_j}{\mu_j}\Big)\|_\infty \|\Lambda^+\|_1 \leq \\ &\|(F(z)^{-1} - P^+)^{-1}\|_\infty^2 \|\|\Lambda^+\|_1 \|(z - y)D_\mu^{-1}\|_\infty \leq \\ &\|(F(z)^{-1} - P^+)^{-1}\|_\infty^2 \|\|\Lambda^+\|_1 \|D_\mu^{-1}\|_\infty \|z - y\|_1 = \\ &L\|z - y\|_\infty \end{aligned} \quad (2.5.6)$$

for a suitable constant $L > 0$, where we used that $(F(z)^{-1} - P^+)^{-1}$ is bounded for $z \geq 0$ ((2.5.4)) and that all norms of $\mathbb{R}^N$ are equivalent. Therefore $G(y)$ is Lipschitz continuous for $y \geq 0$.

According to Brouwer Theorem 1.3.3 applied to function G and to the compact convex set $C = \mathbb{R}_+^N \cap B(0, \delta)$, there exist at least a fixed point

$y = \lambda^- - \Lambda^- \geq 0$, yielding a non-negative solution (λ^-, λ^+) to (2.3.2) by setting $\lambda^- = y + \Lambda^-$ and

$$\lambda^+ = (I - F(y)P^+)^{-1}\Lambda^+. \tag{2.5.7}$$

□

This theorem proves that a nonnegative solution to equations (2.3.2) always exists, but it does not say anything about the stability of the network. If the fixed point of the function G is such that $0 < q_i < 1$ for all $i = 1, \ldots, N$, where $q_i = \min(1, \frac{\lambda_i^+}{\mu_i + \lambda_i^-})$, then the stationary distribution exists unique and it is given by $\pi(\mathbf{k}) = \prod_{i=1}^N \pi(k_i) = \prod_{i=1}^N (1 - q_i) q_i^{k_i}$ (see Theorem 2.3.1).

If, on the other hand, we obtain that $q_i = 1$ for some i, then queue i and consequently the network is unstable and the stationary distribution does not exist.

Chapter 3

Applications

In section 2.1.2 we cited some areas of applications of the G-networks. In this chapter we briefly present the use of G-networks in the context of *supervised learning* and an application in Telecommunication Systems, namely the *Traffic Matrix Estimation* problem ([14]).

3.1 Learning in the G-network

Supervised learning generally refers to the task of inferring a function using a set of *training examples*. In general, each training example is an input-output pair (A, B), where A and B are usually collections of vectors: the function to be inferred, when presented with the input A, must give B as an output.

When using G-networks, we are presented with a set of input-output pairs of the form $\{(X_k, Y_k)\}_{k=1,\dots,K}$, $K \in \mathbb{N}$, where:

- the k-th training input X_k consists of the vectors $\Lambda_k^+ = [\Lambda_{1k}^+, \dots, \Lambda_{Nk}^+]$ and $\Lambda_k^- = [\Lambda_{1k}^-, \dots, \Lambda_{Nk}^-]$, i.e. the rates of the positive and negative customer arrival processes from the outside.

- The k-th training output consists of the vector $Y_k = [y_{1k}, \dots, y_{NK}]$, whose elements $y_{ik} \in (0, 1)$ represent the desidered steady-state occupation probabilities of the queues. Usually only a subset of all the queues is used as *output* queues: that is, queues which only interact with the outside, i.e. for which $d_i = 0$. In the context of supervised learning, we are only interested in the occupation probabilities of the output queues.

Training the G-network means finding the parameters P^+, P^-, μ which minimize a general quadratic function of the form

$$E(\mathbf{q_1}, \ldots, \mathbf{q_K}) = \sum_{k=1}^{K} E_k(\mathbf{q_k}), \tag{3.1.1a}$$

$$E_k(\mathbf{q_k}) = \sum_{i=1}^{N} c_i (f_i(q_{ik}) - y_{ik})^2 \tag{3.1.1b}$$

where:

- $\mathbf{q_k} = [q_{1k}, \ldots, q_{Nk}]$ and $q_{1k}, \ldots, q_{Nk}$ are the steady-state occupation probabilities of the G-network when presented with the outside arrival rates Λ_k^+, Λ_k^-, for $k = 1, \ldots, K$.
- c_i equals 1 if the queue i is an output queue, 0 otherwise.
- $f_i : (0,1) \to \mathbb{R}$ is a differentiable function.

Remark 3.1.1. The function $\mathbf{q_k} = \mathbf{q_k}((\mu_i), (p_{ij}^+), (p_{ij}^-))$ is a function of the inputs rates Λ_k^+, Λ_k^- (which are given by the training set) and of the network parameters $\mu_i, p_{ij}^+, p_{ij}^i$, for $i, j = 1, \ldots, N$. This is because the $q_{1k}, \ldots, q_{Nk}$, for $i = 1, \ldots, N$, $k = 1, \ldots, K$, are the solutions of the traffic equations

$$\begin{cases} \lambda_{ik}^+ = \Lambda_{ik}^+ + \sum_{j=1}^N \mu_j q_j p_{j,i}^+ \\ \lambda_{ik}^- = \Lambda_{ik}^- + \sum_{j=1}^N \mu_j q_j p_{j,i}^- & i = 1, \ldots, N \\ q_i = \min\left(1, \frac{\lambda_{ik}^+}{\mu_i + \lambda_{ik}^-}\right). \end{cases} \tag{3.1.2}$$

Let us introduce the following variables, known as the *weights* of the network:

$$w_{ij}^+ = \mu_i p_{ij}^+ \geq 0, \quad w_{ij}^- = \mu_i p_{ij}^- \geq 0 \qquad \forall\, i, j = 1, \ldots, N \tag{3.1.3}$$

The system (3.1.2) then reads

$$\begin{cases} \lambda_{ik}^+ = \Lambda_{ik}^+ + \sum_{j=1}^N w_{ji}^+ q_j \\ \lambda_{ik}^- = \Lambda_{ik}^- + \sum_{j=1}^N w_{ji}^- q_j & i = 1, \ldots, N \\ q_i = \min\left(1, \frac{\lambda_{ik}^+}{\mu_i + \lambda_{ik}^-}\right). \end{cases} \tag{3.1.4}$$

Observe that when using the variables w_{ij}^+, w_{ij}^-, the condition

$$\sum_{j=1}^{N} (p_{ij}^+ + p_{ij}^-) + d_i = 1 \quad \forall \quad i = 1, \ldots, N \tag{3.1.5}$$

reads

$$\sum_{j=1}^{N}(w_{ij}^{+} + w_{ij}^{-}) = \mu_i(1 - d_i) \quad \forall\, i = 1, \ldots, N \tag{3.1.6}$$

In the context of supervised learning ([6]), usually one wants to use the w_{ij}^{+}, w_{ij}^{-} as the defining parameters of the G-network. In order to do this, one has to fix the d_i and recover the μ_i-s by (3.1.6) and the p_{ij}^{+}, p_{ij}^{-}-s by (3.1.3), for $i, j = 1, \ldots, N$. In the following we will suppose to have done that.

Therefore the function $\mathbf{q_k} = \mathbf{q_k}((w_{ij}^{+}), (w_{ij}^{-}))$, when the $\Lambda_k^{+}, \Lambda_k^{-}$ are given, is actually a function of the weights w_{ij}^{+} and w_{ij}^{-}, $i, j = 1, \ldots, N$.

As a consequence, also the error functions $E_k(\mathbf{q_k}) = E_k(\mathbf{q_k}((w_{ij}^{+}), (w_{ij}^{-})))$, $k = 1, \ldots, K$, when the $\Lambda_k^{+}, \Lambda_k^{-}$ are given, are functions of the weights w_{ij}^{+} and w_{ij}^{-}, $i, j = 1, \ldots, N$.

The optimization problem that we have to solve then reads

$$\begin{cases} \min \sum_{k=1}^{K} E_k(\mathbf{q_k}((w_{ij}^{+}), (w_{ij}^{-}))) \\ \text{s.t. } \; w_{ij}^{+} \geq 0, w_{ij}^{-} \geq 0 \quad \forall\, i, j = 1, \ldots, N \end{cases} \tag{3.1.7}$$

In [6] a gradient descent algorithm to solve (3.1.7) is presented, which we briefly describe.

Let us denote by the generic term w_{uv} either the element w_{uv}^{+} or w_{uv}^{-}, for $u, v = 1, \ldots, N$. The iterative rule for updating the weigths using the kth input-output pair at step $\tau + 1$ of the algorithm is

$$w_{uv}^{\tau+1} = w_{uv}^{\tau} - \eta \Big[\frac{\partial E_k}{\partial w_{u,v}}\Big]_{\tau} \tag{3.1.8}$$

where $\eta > 0$ is the step size, which can be changed during the algorithm, and where the operator $[\cdot]_\tau$ denotes that all calculations are performed using the weight values of step τ and the q_{ik} values derived from solving equation (3.1.2) when the current weights $w_{uv}^{(\tau)}$ are used.

According to 3.1.1, we have

$$\Big[\frac{\partial E_k}{\partial w_{uv}}\Big]_{\tau} = \sum_{i=1}^{N} c_i(f_i(q_{ik}) - y_{ik}) \Big[\frac{\partial f_i}{\partial q_i}\frac{\partial q_i}{\partial w_{uv}}\Big]_{\tau} \tag{3.1.9}$$

Gelenbe showed that

$$\frac{\partial \mathbf{q}}{\partial w_{uv}} := \left[\frac{\partial q_1}{\partial w_{uv}}, \ldots, \frac{\partial q_N}{\partial w_{uv}}\right] = \boldsymbol{\gamma}(u,v)(I - \mathcal{W})^{-1} \quad \forall\, u, v = 1, \ldots, N \tag{3.1.10}$$

where:

- the row vector $\boldsymbol{\gamma}(u,v)$ denotes eiher $\boldsymbol{\gamma}^+(u,v)$ or $\boldsymbol{\gamma}^-(u,v)$, given by

$$\gamma_i^+(u,v) = \begin{cases} -1/D(i) & \text{if} \quad u = i, v \neq i \\ 1/D(i) & \text{if} \quad u \neq i, v = i \\ 0 & \text{otherwise} \end{cases} \tag{3.1.11a}$$

$$\gamma_i^-(u,v) = \begin{cases} -(1+q_i)/D(i) & \text{if} \quad u = i, v = i \\ -1/D(i) & \text{if} \quad u = i, v \neq i \\ -q_i/D(i) & \text{if} \quad u \neq i, v = i \\ 0 & \text{otherwise} \end{cases} \tag{3.1.11b}$$

$$\text{where } D(i) = \mu_i + \Lambda_i^- + \sum_{j=1}^{N} q_j w^-(i,j) \tag{3.1.11c}$$

- $\mathcal{W} \in \mathbb{R}^{N \times N}$ is the matrix with elements $\varpi_{ij} = \frac{(w_{ij}^+ - w_{ij}^-)q_j}{D(j)}$ for $i, j = 1, \ldots, N$.

All this quantities depend on the current values (that is, at step τ) of w_{uv} and $\mathbf{q}_k$. The algorithms then reads:

1) Initialise the weights $w_{ij}^{(0)}$ $\forall i, j$,as well as the step size η.

2) For each successive value of k, starting with $k = 1$, initialise Λ_{ik}^+ and Λ_{ik}^- according to $\mathbf{X}_k$. Then:

2a) Solve the system 3.1.4 using the current weight values.

2b) Based on the values obtained compute the matrix $\mathcal{W}$ and $\boldsymbol{\gamma}(i,j)$ $\quad \forall i, j$.

2c) Calculate $\frac{\partial \mathbf{q}}{\partial w_{ij}}$ according to eq. 3.1.10, for all i, j.

2d) Update the weights w_{ij} using 3.1.8 and 3.1.9. If this yields a negative value for a weight, either set this weight to zero or repeat the step with a smaller η.

2e) Repeat steps $2a), 2b), 2c)$ and $2d)$ until convergence

3) Repeat the procedure of step 2) until convergence.

The computational cost for updating the weights (steps $2b, 2c$ and $2d$) is $O(N^3)$, since the most demanding operation is the inversion of matrix $I - \mathcal{W}$ in eq. 3.1.10.

Our interest for this application resides in step $2a$): it requires the solution of the system 3.1.4, which is the subject matter of this thesis.

3.2 An application: Traffic Matrix estimation in large-scale IP networks

In this section we suggest a real problem that motivates the need for developing efficient methods for the computation of the steady-state distribution in a G-network.

In Telecommunication engineering, knowing the traffic that flows through a large-scale network is a key issue in the design of network. In a IP network this traffic is typically described by a Traffic Matrix (TM). The TM represents the volume of traffic transmitted between every pair of nodes in a network, also referred to as the origin-destination (OD) traffic flows. Directly measuring the TM is often a prohibitive task: as a consequence, network analysis requires efficient TM estimation methods.

Consider a network with l links and m OD flows. Usually $m >> l$. A traffic matrix is formally described by

- a vector $X_t = [x_t(1), \ldots, x_t(m)]^\top \in \mathbb{R}^m$, where t is a variable representing time and $x_t(i)$ is the traffic volume of the i-th OD flow, for $i = 1, \ldots, m$.
- a vector $Y_t = [y_t(1), \ldots, y_t(l)]^\top \in \mathbb{R}^l$, where $y_t(j)$ is the aggregated traffic volume in link j, $j = 1, \ldots, j$, that is the traffic originated by all the OD flows that traverse the link j.
- a *routing* matrix $R \in \mathbb{R}^{l \times m}$, where

$$R_{ij} = \begin{cases} 1 & \text{if the OD flow } j \text{ traverses link } i. \\ 0 & \text{otherwise.} \end{cases}$$

The relation between X_t and Y_t reads

$$Y_t = RX_t \tag{3.2.1}$$

While it is possibile to efficiently measure the aggregated data Y_t using well-known protocols, the exact measurement of X_t is in general too costly. The TM estimation problem consists in computing the vector X_t form the known vector Y_t.

A lot of approaches to tackle the TM estimation problem have been investigated in the literature. In this thesis we are interested in the approach by P. Casas and S. Vaton ([14]), which makes use of Random Neural Networks (i.e., G-networks): this method involve the training of several of G-networks of moderate size, and thus requires the solution of the system (2.3.1) a large number of times.

For each OD flow k, $k = 1, \ldots, m$, let n_k be the number of links it crosses, and let $\delta_k = [\delta_k(1), \ldots, \delta_k(n_k)]^\top \in \mathbb{R}^{n_k}$ contain the indexes of the n_k non-zero elements in the k-th column of R, i.e the indexes of the link crossed by OD flow k. We want to find functions $f_k : \mathbb{R}^{n_k} \to \mathbb{R}$, $k = 1, \ldots, m$, such that

$$x_t(k) = f_k(Y_t(\delta_k))$$

where $Y_t(\delta_k) := y_t(\delta_1(1), \ldots, \delta_k(n_k))$ is the vector containing the traffic volume of the n_k links that are crossed by the k-th OD flow. Intuitively, the functions f_k *extract* the volume of OD flow k from the trace that this flow leaves in the n_k links it traverses.

In [14] the authors propose to approximate each functions f_k, for $k = 1, \ldots, m$, with a G-network, resulting in a total of m G-networks to be trained. Following a classic neural network approach, they use particular *feed-forward* networks (section 2.5.1) with a three-layer structure (fig. 3.1)

Three-layers network have a particular simple structure, and are composed by input, hidden and outpute queues. *Input* queues (first layer) receive only positive customers from the outside and send customers to *hidden* queues (second layer). Hidden queues do not interact with the outside and only send customers to *output* queues. Output queues only send customers to the outside. If the queues are indexed as

$$\underbrace{i_1, \ldots, i_I}_{\text{input}}, \underbrace{i_{I+1}, \ldots, i_{I+H}}_{\text{hidden}}, \underbrace{i_{I+H+1}, \ldots, i_{I+H+O}}_{\text{output}}$$

where I, H, O are the number of respectively input, hidden and output queues, the outsided arrival rates are such that

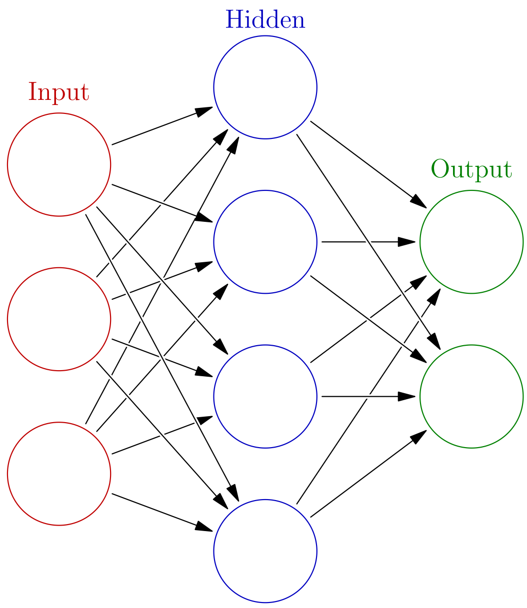

Figure 3.1: An example of a three-layers feed-forward network. Image from Wikipedia.

- $\Lambda^+(i_v) = 0 \quad \forall\, v = I+1, I+H+O$, i.e., positive customer form the outside can arrive only at input queues.

- $\Lambda^-(i_v) = 0 \quad \forall\, v = 1, \ldots, I+H+O$, i.e. there are no negative customers arriving from the outside.

For the k-th G-network, $k = 1, \ldots, m$, the authors set $I_k = n_k$ input neurons, H_k hidden neurons (which is not a priori fixed and is set through euristical methods), and $O_k = 1$ output neuron. They perform numerical experiments using data form the real network *Abilene*, an Internet2 backbone network at the US. For this particular application, there are $m = 132$ OD flows, $l = 30$ links, and the other parameter read $I_k \geq 4$, $4 \leq H_k \leq 9$ and $O_k = 1$ for all $k = 1, \ldots, m$.

For each $k = 1, \ldots, m$, they train the k-th G-network using the *training dataset*

$$\{Y_t(\delta_k), x_t(k)\}_{t \in T_{\text{learn}}}$$

where $T_{\text{learn}} = \{t_1, \ldots, T_{288}\}$, composed of 24 hours of direct OD flows measurement. The learning algorithm used is described in section 3.1.

Then, a *validation dataset*

$$\{Y_t(\delta_k), x_t(k)\}_{t \in T_{\text{val}}}$$

is used to verify if the m trained G-networks yield a good approximation of the unknown functions f_k, $k = 1, \ldots, m$, where $T_{\text{val}} = \{t_1, \ldots, T_{2016}\}$, composed of a week of direct OD flows measurements. As a global indication of the accuracy of the estimation, the relative root mean squared error is used:

$$\text{RMMSE}(t) = \frac{\sqrt{\sum_{k=1}^{m}(x_t(k) - q_t(k))^2}}{\sqrt{\sum_{k=1}^{m} x_t(k)^2}}, \qquad \forall\, t \in T_{\text{val}} \tag{3.2.2}$$

where $q_t(k)$ it the output of the k-th trained G-network when presented with the input $Y_t(\delta_k)$. The authors in [14] showed that the RMMSE(t) is below 0.08 for more than 90% of the $t \in T_{\text{val}}$, with a mean value of 0.0422%, resulting in a huge improvement with respect to other TM estimation methods.

In this simple three-layer feed-forward G-networks scenario, the networks have really small dimensions, ranging from 9 to 14, and there are 132 of them. Each G-network is trained with a learning dataset composed of 288 input-output pairs, and according to the gradient descent methods described in 3.1 this implies that the system (2.3.2) must be solved a number of times proportional to $132 \cdot 288 = 38016$. In this case, due to the feed-forward structure, no particular algorithm is required for the solution of the correspondent systems (2.3.1).

However, when using G-networks with more complex topologies, solving the equations (2.3.2) becomes a non-trivial task, resulting in the need for efficient algorithms. On the basis of this observation, the rest of the thesis is devoted to the study of new numerical methods for the computation of the steady-state distribution.

Chapter 4

Numerical methods for computing the steady-state distribution

As shown in section 2.5.3, a non-negative solution $\{\lambda_i^+, \lambda_i^-\}_{i=1}^N$ of the nonlinear system (2.3.2) always exists, but its direct computation is possibile only in the particular case of feed-forward networks, described in section 2.5.1. In the general case, we have shown that solving equations (2.3.2) is equivalent to finding the fixed point of the function $G(y) = \Lambda^+(I - F(y)P^+)^{-1}F(y)P^-$, for which iterative methods are needed.

In section 4.1 we will describe an algorithm developed by Fourneau in 1991 ([4]), while in section 4.2 two new numerical methods will be presented and studied.

4.1 Fourneau iteration.

In [4], J.M. Fourneau developed an algorithm to compute the steady state distribution of a standard G-network. Recall that the steady state distribution is obtained by solving the non-linear system of equations (2.3.2) with unknowns λ_i^+, λ_i^-, which read

$$\lambda_i^+ = \Lambda_i^+ + \sum_{j=1}^N \mu_j q_j p_{j,i}^+$$

$$\lambda_i^- = \Lambda_i^- + \sum_{j=1}^N \mu_j q_j p_{j,i}^-$$

with

$$q_i = \min\Big(1, \frac{\lambda_i^+}{\mu_i + \lambda_i^-}\Big).$$

Fourneau considers, for all index $i = 1, \dots, N$, six sequences of real numbers

$$(\overline{q_i})_{k\geq 0}, (\underline{q_i})_{k\geq 0}, (\overline{\lambda_i^+})_{k\geq 0}, (\underline{\lambda_i^+})_{k\geq 0}, (\overline{\lambda_i^-})_{k\geq 0}, (\underline{\lambda_i^-})_{k\geq 0}$$

defined by induction on $k \geq 0$ as follows:

$$(\overline{\lambda_i^+})_k = \Lambda_i^+ + \sum_{j=1}^{N} \mu_j p_{j,i}^+ (\overline{q_j})_k \tag{4.1.2a}$$

$$(\underline{\lambda_i^+})_k = \Lambda_i^+ + \sum_{j=1}^{N} \mu_j p_{j,i}^+ (\underline{q_j})_k \tag{4.1.2b}$$

$$(\overline{\lambda_i^-})_k = \Lambda_i^- + \sum_{j=1}^{N} \mu_j p_{j,i}^- (\overline{q_j})_k \tag{4.1.2c}$$

$$(\underline{\lambda_i^-})_k = \Lambda_i^- + \sum_{j=1}^{N} \mu_j p_{j,i}^- (\underline{q_j})_k \tag{4.1.2d}$$

$$(\overline{q_i})_{k+1} = \min(1, (\overline{\lambda_i^+})_k / \mu_i + (\underline{\lambda_i^-})_k) \tag{4.1.2e}$$

$$(\underline{q_i})_{k+1} = \min(1, (\underline{\lambda_i^+})_k / \mu_i + (\overline{\lambda_i^-})_k) \tag{4.1.2f}$$

with the following initial values for the sequences $(\overline{q_i})_{k\geq 0}$ and $(\underline{q_i})_{k\geq 0}$:

$$(\overline{q_i})_0 = 1, \quad (\underline{q_i})_0 = 0 \tag{4.1.3}$$

For each $k \geq 0$, starting from $k = 0$, the iteration proceeds as follows: first compute the $(\overline{\lambda_i^+})_k, (\underline{\lambda_i^+})_k, (\overline{\lambda_i^-})_k, (\underline{\lambda_i^-})_k$ from the equations (4.1.2), which only depends on the kwnon values $(\overline{q_i})_k$ and $(\overline{q_i})_k$, then compute $(\overline{q_i})_{k+1}$ and $(\overline{q_i})_{k+1}$ using the just computed values.

Theorem 4.1.1 ([4]). *If for any $i = 1, \dots, N$ one of the following assumption is satisfied:*

- *there is a strictly positive probability that a positive customer leaves the queue to go outside, i.e. $d_i > 0$.*

- *there is a strictly positive probability that a customer, either positive or negative, joins a queue j where the rate of negative customers coming from the outside is strictly positive, i.e.* $p_{ij}^{+} + p_{ij}^{-} > 0$ *and* $\Lambda_j^{-} > 0$.

then the algorithm (4.1.2) *converges to a solution of the system* (2.3.2).

The complete proof can be found in [4]. Fourneau proved that the sequences $(\overline{q_i})_k$ and $(\underline{q_i})_k$ are respectively upper and lower bounds for the q_i and that

$$\sum_{i=1}^{N} \mu_i[(\overline{q_i})_{k+1} - (\underline{q_i})_{k+1}] \leq \epsilon \sum_{i=1}^{N} \mu_i[(\overline{q_i})_k - (\underline{q_i})_k] \tag{4.1.4}$$

where

$$\epsilon_i = \sum_{j=1}^{N} (p_{ij}^{+} + p_{ij}^{-}) \frac{\mu_j}{\mu_j + \Lambda_j^{-}}, \quad \epsilon = \max_{i=1,\ldots,N} \epsilon_i < 1.$$

Therefore the sequence $\sum_{i=1}^{N} \mu_i[(\overline{q_i})_k - (\underline{q_i})_k]$ converges linearly to zero, proving the thesis.

If the solution computed by the Fourneau algorithm is such that all the components q_i are smaller than 1, then the network is stable and the network stationary distribution exists and is given by (2.3.4).

Otherwise if some q_i are equal to 1, the network is unstable and the stationary distribution does not exist.

4.1.1 Complexity

Each iteration of the algorithm consists in the computations of the new values of the sequences (4.1.2) and the new values of the difference $(\overline{q_i})_k - (\underline{q_i})_k$ to check the accuracy. There are $6N$ new values to compute at each iteration, four of which requiring $O(N)$ time, while two of them requiring $O(1)$ time. Therefore the complexity of each iteration is $O(N^2)$. In conclusion, the Fourneau algorithm is an iterative methods with linear rate of convergence requiring $O(N^2)$ time per step.

In the rest of this thesis we will develop two new iterative methods having respectively linear and quadratically rate of convergence, both requiring $O(N^3)$ time per step. The second method, in particular, will be an attractive alternative to the Fourneau iteration especially for G-network with a moderate number of queues.

4.2 New numerical methods

In this section we will present a new formulation of the fixed point equation $y = G(y)$, which will allow us to introduce two new iterative methods for the solution of the traffic equations (2.3.2).

Recall that

$$G(y) = \Lambda^+(I - F(y)P^+)^{-1}F(y)P^-$$
$$F(y) = \mathrm{diag}((f_j(y)))$$

with $y = (y_j) \geq 0$, $f_j(y) = \frac{\mu_j}{\mu_j+\lambda_i^-} = \frac{\mu_j}{\mu_j+\Lambda_j^-+y_j} > 0$, Λ^+, Λ^- are nonnegative row vectors, μ is a positive row vector and P^+ and P^- are nonnegative irriducible matrices such that $P^+ + P^-$ is substochastic, as described in section 2.1.1.

Since $f_j(y) > 0 \quad \forall\, j = 1\ldots, N$, the diagonal matrix $F(y)$ is non-singular and we can write $G(y) = \Lambda^+(F(y)^{-1} - P^+)^{-1}P^-$.

Set $D_z = F(y)^{-1}$, so that $D_z = \mathrm{diag}(z)$ and

$$z_j = \frac{1}{f_j(y)} = \frac{\mu_j + \lambda_j^-}{\mu_j} = \frac{\mu_j + \Lambda_j^- + y_j}{\mu_j} = \frac{\alpha_j + y_j}{\mu_j} \geq 1. \tag{4.2.2}$$

Set also $\alpha_j = \mu_j + \Lambda_j^- > 0$, $\alpha = (\alpha_j)$ and $D_\mu = \mathrm{diag}(\mu)$, so that the relation between variables y and z is $y = zD_\mu - \alpha$.

The equation $y = G(y)$ can therefore be rewritten equivalently as

$$zD_\mu - \alpha = \Lambda^+(D_z - P^+)^{-1}P^-$$

yielding the new fixed point formulation

$$\begin{cases} z = T(z) \\ T(z) = \Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1} \end{cases} \tag{4.2.3}$$

where the function $T(z)$ is defined for $z \geq \mathbf{1}$. The variables y and z and the functions G and T satisfy the relations

$$y = zD_\mu - \alpha \tag{4.2.4a}$$
$$G(y) = T(z)D_\mu - \alpha \tag{4.2.4b}$$

Remark 4.2.1. The result of theorem 2.5.3 immediately translate in this new formulation, yielding the existence of a fixed point $z^* = (y^* + \alpha)D_\mu^{-1} \geq \mathbf{1}$ for the function $T(z)$ in $z \geq \mathbf{1}$.

Remark 4.2.2. For any non-negative irreducible matrix P with $\rho(P) < 1$ and for any $z \geq \mathbf{1}$, $D_z - P$ is a nonsingular M-matrix:

- It is a Z-matrix, since D_z is diagonal and P is non-negative.
- It is non-singular and its inverse is non-negative.

  Indeed, $D_z - P = D_z(I - D_z^{-1}P)$, $D_z^{-1}P$ is irreducible and $0 \leq D_z^{-1}P \leq P$, yielding that $\rho(D_z^{-1}P) \leq \rho(P) < 1$ by the Perron-Frobenius theorem 1.2.2. Therefore $D_z - P$ is invertible and $(D_z - P)^{-1} = (I - D_z^{-1}P)^{-1}D_z^{-1} = \sum_{n\geq 0}(D_z^{-1}P)^n D_z^{-1} \geq 0$.

By the characterization of non-singular M-matrices 1.2.3, we get the thesis.

In particular, matrices $D_z - P^+$, $D_z - P^-$ and $D_z - P^+ - P^-$ are non-singular M-matrices.

Remark 4.2.3. If z^* is a fixed point of T, yielding a non-negative solution $\{\lambda_i^+, \lambda_i^-\}_{i=1}^N$ of equations (2.3.2), the stability condition $q_i = \frac{\lambda_i^+}{\mu_i + \lambda_i^-} < 1$ for all $i = 1, \ldots, N$ can we rewritten as

$$\Lambda^+(D_z^* - P^+)^{-1}D_\mu^{-1} < \mathbf{1}. \qquad (4.2.5)$$

Indeed, since $z_i = \frac{\mu_i + \lambda_i^-}{\mu_i}$ from (4.2.2), we have $q_i = \frac{\lambda_i^+}{\mu_i + \lambda_i^-} = \frac{\lambda_i^+}{z_i \mu_i}$. Recall that

$$\lambda^+ = \Lambda^+(I - F(y)P^+)^{-1} = \Lambda^+(I - D_z^{-1}P^+)^{-1}$$

(see (2.5.7)), yielding

$$q = \Lambda^+(I - D_z^{-1}P^+)^{-1}D_z^{-1}D_\mu^{-1} = \Lambda^+(D_z - P^+)^{-1}D_\mu^{-1}. \qquad (4.2.6)$$

In section 4.2.1 we will study the properties of function $T(z) = \Lambda^+(D_z - P^+)^{-1}P^- D_\mu^{-1} + \alpha D_\mu^{-1}$. The advantage of this new formulation, with respect to the previous formulation $y = G(y)$, is that we will be able to prove that T is Fréchet-differentiable and to easily compute its Fréchet derivative. In sections 4.2.2 and 4.2.3 these properties will be used to prove the convergence of two new iterative methods for the solution of $z = T(z)$, namely a fixed point iteration and a Newton-Raphson method.

4.2.1 Properties

Lemma 4.2.1. *The function* $T(z) = \Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1}$*, for* $z \geq \mathbf{1}$*, is component-wise nonnegative and non increasing. Moreover, it is Lipschitz continuous and bounded.*

Proof. By remark 4.2.2 we have that $D_z - P^+$ is a nonsingular M-matrix, therefore $(D_z - P^+)^{-1} \geq 0$ by theorem 1.2.2. Since $\Lambda^+, P^-, \alpha, D_\mu^{-1} \geq 0$, we get that $T(z) = \Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1} \geq 0$. The diagonal matrix $D_z = \mathrm{diag}(z)$ is non-decreasing in z, yielding that, for $z \geq \mathbf{1}$, $(D_z - P^+)^{-1} = (I - D_z^{-1}P^+)^{-1}D_z^{-1} = \sum_{n\geq 0}(D_z^{-1}P^+)^n D_z^{-1}$ and $T(z) = \Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1}$ are non-increasing, since Λ^+, P^+, P^- are non-negative. As a consequence

$$(D_z - P^+)^{-1} \leq (I - P^+)^{-1} \qquad \forall z \geq \mathbf{1} \tag{4.2.7}$$

yielding that $(D_z - P^+)^{-1}$ is bounded for $z \geq \mathbf{1}$.

Regarding the Lipscthiz continuity, for all $z, w \in \mathbb{R}^N$,$z, w \geq 1$ we have

$$\begin{aligned} T(z) - T(w) =& \Lambda^+\Big[(D_z - P^+)^{-1} - (D_w - P^+)^{-1}\Big]P^-D_\mu^{-1} \\ & \Lambda^+(D_w - P^+)^{-1}(D_w - D_z)(D_z - P^+)^{-1}P^-D_\mu^{-1} = \\ & \mathbf{1}\,\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})\mathrm{diag}(w - z)(D_z - P^+)^{-1}P^-D_\mu^{-1} \end{aligned} \tag{4.2.8}$$

yielding, with similar passages as in (2.5.6), $\|T(z) - T(w)\|_\infty \leq K\|z - w\|_\infty$ for and a suitable costant K, since $(D_z - P^+)^{-1}$ is bounded for $z \geq 1$.

Finally, by (4.2.4) and (2.5.4) we have $T(z) = (G(y) + \alpha)D_\mu^{-1}$ and $0 \leq G(y) \leq \Lambda^+(I - P^+)^{-1}P^-$ for all $y \in (\mathbb{R}^N)_+$, yielding that $T(z)$ is bounded for $z \geq \mathbf{1}$.

□

Remark 4.2.4. Equation (4.2.8) in particular implies that for any $w \geq \mathbf{1}$

$$\lim_{\substack{z\to w\\ z\geq \mathbf{1}}} \|(D_z - P^+)^{-1} - (D_w - P^+)^{-1}\| = 0 \tag{4.2.9}$$

since $(D_z - P^+)^{-1}$ is bounded for all $z \geq \mathbf{1}$.

Lemma 4.2.2. *The function* $T(z) = \Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1}$ *is Fréchet differentiable for* $z \geq \mathbf{1}$ *and its Fréchet derivative is*

$$J_T(z) := -\mathrm{diag}(\Lambda^+(D_z - P^+)^{-1})(D_z - P^+)^{-1}P^-D_\mu^{-1}$$

Proof. Taking $z = w + h$ in (4.2.8), for $h \in \mathbb{R}^N$, yields

$$\begin{aligned} T(w+h) - T(w) &= \mathbf{1}\,\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})\mathrm{diag}(-h)(D_{w+h} - P^+)^{-1}P^- D_\mu^{-1} \\ &= -\mathbf{1}\,\mathrm{diag}(h)\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})(D_{w+h} - P^+)^{-1}P^- D_\mu^{-1}, \end{aligned}$$

where we used that diagonal matrices commute. Therefore

$$\lim_{h\to 0} \frac{\|T(w+h) - T(w) - hJ_T(w)\|_\infty}{\|h\|_\infty} =$$
$$\lim_{h\to 0} \frac{\|h\,\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})\Big[(D_{w+h} - P^+)^{-1} - (D_w - P^+)^{-1}\Big]P^- D_\mu^{-1}\|_\infty}{\|h\|_\infty} \le$$
$$\lim_{h\to 0} \|\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})\|_\infty \|P^- D_\mu^{-1}\|_\infty \|(D_{w+h} - P^+)^{-1} - (D_w - P^+)^{-1}\|_\infty = 0$$

since:

- $\|\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})\|_\infty = \|(D_w - P^+)^{-\top}(\Lambda^+)^\top)\|_\infty \le \|(D_w - P^+)^{-1})\|_1 \|\Lambda\|_\infty$, which is bounded for $z \ge \mathbf{1}$ thanks to (4.2.7)
- $\lim_{h\to 0} \|(D_{w+h} - P^+)^{-1} - (D_w - P^+)^{-1}\| = 0$ by (4.2.9).

□

The most important result of this section is the next theorem, which will allow us to study the convergence properties of the computational methods described in the next sections.

Theorem 4.2.1. *Let $z \ge \mathbf{1}$ be such that*

$$\Lambda^+(D_z - P^+)^{-1}D_\mu^{-1} < \mathbf{1}.$$

Then $\rho(J_T(z)) < 1$.

Proof. The idea of the proof is to write $-J_T(z) = M^{-1}N$, with matrices M and N such that M is nonsingular, $M^{-1} \ge 0$,$N \ge 0$ and $M - N$ is a nonsingular M-matrix. Thanks to theorems 1.2.3 and 1.2.4, this will imply that $\rho(M^{-1}N) = \rho(-J_T(z)) = \rho(J_T(z)) < 1$.

Letting $\Delta := \Lambda^+(D_{z^*} - P^+)^{-1}$, the stability condition reads

$$\frac{1}{\Delta_j} > \frac{1}{\mu_j} \tag{4.2.12}$$

for all $j = 1, \ldots, N$.

Write

$$\begin{aligned} -J_T(z) =& \text{diag}(\Lambda^+(D_z - P^+)^{-1})(D_z - P^+)^{-1}P^-D_\mu^{-1} = \\ =& \text{diag}(\Delta)(D_z - P^+)^{-1}P^-D_\mu^{-1} = M^{-1}N \end{aligned}$$

with $M = \text{diag}((\frac{1}{\Delta_j})) \geq 0$ and $N = (D_z - P^+)^{-1}P^-D_\mu^{-1} \geq 0$.

The matrix $M - N = \text{diag}((1/\Delta_j)) - (D_z - P^+)^{-1}P^-D_\mu^{-1}$ is a Z-matrix, since M is diagonal and $N \geq 0$.

Using (4.2.12) yields

$$\begin{aligned} M - N =& \text{diag}((1/\Delta_j)) - (D_z - P^+)^{-1}P^-D_\mu^{-1} \geq \\ =& D_\mu^{-1} - (D_z - P^+)^{-1}P^-D_\mu^{-1} = \\ =& (D_z - P^+)^{-1}(D_z - P^+ - P^-)D_\mu^{-1} \end{aligned}$$

Now, by remark 4.2.2, $D_z - P^+$ and $D_z - P^+ - P^-$ are non-singular M-matrices. Therefore by lemma 1.2.3 there exist a vector $x > 0$ such that $y := (D_z - P^+ - P^-)x > 0$.

Since $\mu > 0$ we have that $\tilde{x} = D_\mu x > 0$. Moreover, since $(D_z - P^+)^{-1} \geq 0$, we have that $(D_z - P^+)^{-1}y > 0$, since if a component of $(D_z - P^+)^{-1}y$ were zero this would imply that the matrix $(D_z - P^+)^{-1}$ has a null row, contradicting the fact that it is non-singular. Therefore

$$\begin{aligned} (M - N)\tilde{x} &\geq (D_z - P^+)^{-1}(D_z - P^+ - P^-)D_\mu^{-1}\tilde{x} \geq \\ &\geq (D_z - P^+)^{-1}(D_z - P^+ - P^-)x = \\ &= (D_z - P^+)^{-1}y > 0 \end{aligned}$$

yielding that $M - N$ is a nonsigular M-matrix, by using lemma 1.2.3 again. □

Recall (4.2.3) that the condition for a G-network to be stable is $\Lambda^+(D_{z^*} - P^+)^{-1}D_\mu^{-1} < \mathbf{1}$, where z^* is the fixed point of the function $T(z)$ for $z \geq \mathbf{1}$. We then have the following corollary.

Corollary 4.2.1. *Suppose that the G-network is stable, and let $z^* \geq \mathbf{1}$ be the fixed point of the function $T(z)$ for $z \geq \mathbf{1}$, which exists by theorem 2.5.3. Then $\rho(J_T(z^*)) < 1$.*

Lemma 4.2.3. *The function $J_T(z)$ is Lipschitz continuous for $z \geq \mathbf{1}$.*

Proof. Recall that $J_T(z) = -\text{diag}(\Lambda^+(D_z - P^+)^{-1})(D_z - P^+)^{-1}P^- D_\mu^{-1}$. Setting $A_z = (D_z - P^+)^{-1}$, by equation (4.2.8) we get $A_{z+h} - A_z = -A_z D_h A_{z+h}$, yielding

$$\begin{aligned}
&(J_T(z) - J_T(z+h)) = \\
&= \Big(\text{diag}(\Lambda^+(D_{z+h} - P^+)^{-1})(D_{z+h} - P^+)^{-1} - \text{diag}(\Lambda^+(D_z - P^+)^{-1})(D_z - P^+)^{-1}\Big) P^- D_\mu^{-1} = \\
&= \Big(\text{diag}(\Lambda^+(A_z - A_z D_h A_{z+h}))(A_z - A_z D_h A_{z+h}) - \text{diag}(\Lambda^+ A_z) A_z\Big) P^- D_\mu^{-1} = \\
&= -\Big(\text{diag}(\Lambda^+ A_z)(A_z D_h A_{z+h}) - \text{diag}(\Lambda^+ A_z D_h A_{z+h}) A_z + \\
&\quad + \text{diag}(\Lambda^+ A_z D_h A_{z+h})(A_z D_h A_{z+h})\Big) P^- D_\mu^{-1} = \\
&= -\Big(\text{diag}(\Lambda^+ A_{z+h})(A_z D_h A_{z+h}) - \text{diag}(\Lambda^+ A_z D_h A_{z+h}) A_z\Big) P^- D_\mu^{-1}.
\end{aligned} \tag{4.2.16}$$

Using norms we get

$$\begin{aligned}
&\|(J_T(z) - J_T(z+h))\|_\infty = \\
&\|\Big(-\text{diag}(\Lambda^+ A_{z+h})(A_z D_h A_{z+h}) - \text{diag}(\Lambda^+ A_z D_h A_{z+h}) A_z\Big) P^- D_\mu^{-1}\|_\infty \leq \\
&\Big(\|\text{diag}(\Lambda^+ A_{z+h})\|_\infty \|A_z D_h A_{z+h}\|_\infty + \|\text{diag}(\Lambda^+ A_z D_h A_{z+h})\|_\infty \|A_z\|_\infty\Big) \|P^- D_\mu^{-1}\|_\infty = \\
&\Big(\|A_{z+h}^\top (\Lambda^+)^\top\|_\infty \|A_z D_h A_{z+h}\|_\infty + \|(A_z D_h A_{z+h})^\top (\Lambda^+)^\top\|_\infty \|A_z\|_\infty\Big) \|P^- D_\mu^{-1}\|_\infty \leq \\
&\Big(\|A_{z+h}^\top\|_\infty \|(\Lambda^+)^\top\|_\infty \|A_z\|_\infty \|A_{z+h}\|_\infty + \|A_z^\top\|_\infty \|A_{z+h}^\top\|_\infty \|A_z\|_\infty \|(\Lambda^+)^\top\|_\infty\Big) \|P^- D_\mu^{-1}\|_\infty \|D_h\|_\infty \leq \\
&\Big(\|A_{z+h}\|_1 \|\Lambda^+\|_\infty \|A_z\|_\infty \|A_{z+h}\|_\infty + \|A_z\|_1 \|A_{z+h}\|_1 \|A_z\|_\infty \|\Lambda^+\|_\infty\Big) \|P^- D_\mu^{-1}\|_\infty \|D_h\|_\infty \leq \\
&C\|D_h\|_\infty = C\|h\|_\infty.
\end{aligned}$$

where we used that A_w is bounded for all $w \geq \mathbf{1}$ ((4.2.7)) and we implicitly transposed the vectors defining the diagonal matrices.

□

4.2.2 Fixed-point iteration

In the following we will assume that the G-network is stable, that is $\Lambda^+(D_{z^*} - P^+)^{-1} D_\mu^{-1} < \mathbf{1}$.

The first method we consider for solving the equation $z = T(z)$ is the following fixed point iteration:

$$\begin{cases} z^{(k+1)} = T(z^{(k)}) & k \geq 0 \\ z^{(0)} \in \mathbb{R}^N, \ z^{(0)} \geq \mathbf{1}. \end{cases} \tag{4.2.18}$$

Remark 4.2.5. Since the function $T(z)$ is defined for $z \geq \mathbf{1}$, we must ensure that the iteration (4.2.18) is well defined, i.e. that $T(z^{(k)}) \geq \mathbf{1}$ for all $k \geq 0$. Indeed, if $z \geq \mathbf{1}$ we have

$$T(z) = \Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1} \geq \alpha D_\mu^{-1} = \mathbf{1} + \Lambda^- D_\mu^{-1} \geq \mathbf{1},$$

since $\alpha = \mu + \Lambda^-$, $\Lambda^- \geq 0, D_\mu^{-1} \geq 0$ and also $\Lambda^+(D_z - P^+)^{-1}P^-D_\mu^{-1} \geq 0$.

Moreover $T(z) \neq \mathbf{1}$, since $P^-, D_\mu^{-1}, (D_z - P^+)^{-1} \neq 0$ and at least one between Λ^+ and Λ^- is different than zero. Therefore, starting from $z^{(0)} \geq \mathbf{1}$, the iteration (4.2.18) is well defined and $\{z^{(k)}\}_{k\geq 0} \subseteq \{z \in \mathbb{R}^N : z \geq \mathbf{1}, z \neq \mathbf{1}\}$. As a consequence, also for the fixed point z^* it holds that $z^* \geq \mathbf{1}, z \neq \mathbf{1}$.

We are now able to prove that the iteration (4.2.18) is locally convergent:

Theorem 4.2.2. *Suppose that the G-network is stable. Then:*

i) the function $T : \mathbb{R}^N \to \mathbb{R}^N$ is contractive in a neighbourhood of z^*,i.e. there exists a norm $\| \cdot \|$ on $\mathbb{R}^N$, a neighbourhood $I(z^*) \subset \{z \in \mathbb{R}^N : z \geq \mathbf{1}\}$ of z^* and a scalar $0 \leq \gamma < 1$ such that for all $z \in I(z^*)$

$$\|T(z) - T(z^*)\| \leq \gamma \|z - z^*.\|$$

ii) for all $z^{(0)} \in I(z^*)$, the iteration (4.2.18) *converge to z^*, which is the unique fixed point of T in $I(z^*)$.*

Proof. i) Since $\rho(J_T(z^*)) < 1$ (theorem 4.2.1), by lemma 1.2.1 there exist a norm $\| \cdot \|$ on $\mathbb{R}^N$ such that the induced matrix norm satisfies $\|J_T(z^*)\| < 1$. Since $z^* \geq \mathbf{1}$, by continuity of J_T (lemma 4.2.3) there exists a compact neighbourhood of z^* $I(z^*) = \overline{B(z^*, \epsilon)} \subset \{z \in \mathbb{R}^N : z \geq \mathbf{1}\}$, for a certain $\epsilon > 0$, such that $\|J_T(z)\| < 1 \quad \forall z \in I(z^*)$, so that $\gamma := \sup_{z \in I(z^*)} \|J_T(z)\| < 1$. By using the mean value theorem (theorem 1.3.1) we obtain the thesis.

ii) Straightforward application of the Conctraction-Mapping Theorem (theorem 1.3.2).

□

Remark 4.2.6. In lemma 4.2.1 we have shown that $T(z) \leq T(\mathbf{1})$ for all $z \geq \mathbf{1}$, since the function T is non-increasing. Restricting to the neighbourhood $I(z^*) \subset \{z \in \mathbb{R}^N : z \geq \mathbf{1}\}$ of z^* defined in theorem 4.2.2, and considering a square sub-neighbourhood $\widetilde{I}(z^*) := \overline{B(z^*, \tilde{\epsilon})}_{\|\cdot\|_\infty} \subset I(z^*)$, which exists thanks to the equivalence of the norms on $\mathbb{R}^N$, we set

$$z_{\min} := z^* - \tilde{\epsilon} \cdot \mathbf{1} \in \widetilde{I}(z^*),$$

so that for all $z \in \tilde{I}(z^*)$ we have $z \geq z_{\min} \geq \mathbf{1}$. Then $T(z) \leq T(z_{\min})$ for all $z \in \widetilde{I}(z^*)$, since T is non-increasing in $z \geq \mathbf{1}$. In other words, $z_{\min}$ fulfills the role of $\mathbf{1}$ locally.

Let now study the convergence rate of iteration (4.2.18) As shown in the proof of lemma 4.2.1, the function T satisfies, for $z, w \geq \mathbf{1}$,

$$T(z) - T(w) = \mathbf{1}\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})\mathrm{diag}(w - z)(D_z - P^+)^{-1}P^-D_\mu^{-1}$$

i.e.

$$T(z) - T(w) = (w - z)\mathrm{diag}(\Lambda^+(D_w - P^+)^{-1})(D_z - P^+)^{-1}P^-D_\mu^{-1} \quad (4.2.19)$$

where we used that diagonal matrices commute. This relation allows us to prove interesting properties of the sequence (4.2.18).

Lemma 4.2.4. *For the iteration* (4.2.18) *it holds that:*

i) for all $k \geq 1$ *let* $H_k = -diag(\Lambda^+(D_{z^*} - P^+)^{-1})(D_{z^{(k)}} - P^+)^{-1}P^-D_\mu^{-1}$. *Then*
$$z^{(k)} - z^* = (z^{(0)} - z^*)H_0H_1 \cdots H_{k-1} \quad (4.2.20)$$
Moreover, if $z^{(0)} \in I(z^*)$*, where* $I(z^*)$ *is defined in theorem 4.2.2, we have*
$$\lim_{k\to\infty} H_k = J_T(z^*).$$

ii) the sequence $(z^{(k)} - z^*)_{k\geq 0}$ *has alternating sign, i.e. for all* $k \geq 0$ *it holds that*
$$z^{(k)} - z^* \geq 0 \iff z^{(k+1)} - z^* \leq 0 \quad (4.2.21)$$

iii) if $z^{(0)} = T(z_{\min}) \in \tilde{I}(z^*) \subset I(z^*)$*, where* $z_{\min}$ *and* $\tilde{I}(z^*)$ *are defined in remark 4.2.6, the subsequences of odd and even indexes satisfy*

$$\mathbf{1} \leq z_{\min} \leq z^{(2k+1)} \leq z^{(2(k+1)+1)} \leq z^* \leq z^{2(k+1)} \leq z^{(2k)} \qquad \forall k \geq 0 \quad (4.2.22)$$

Proof. i) For all $k \geq 1$, setting $z = z^{(k)}$ and $w = z^*$ in (4.2.19) yields

$$\begin{aligned}
z^{(k)} - z^* =& T(z^{(k-1)}) - T(z^*) = \\
=& (z^* - z^{(k-1)}) \operatorname{diag}(\Lambda^+(D_{z^*} - P^+)^{-1})(D_{z^{(k-1)}} - P^+)^{-1} P^- D_\mu^{-1} = \\
=& (z^{(k-1)} - z^*) H_{k-1} = \\
=& (z^{(k-2)} - z^*) H_{k-2} H_{k-1} = \\
=& \cdots\cdots\cdots\cdots\cdots = \\
=& (z^{(0)} - z^*) H_0 H_1 \cdots H_{k-1}
\end{aligned}$$

Moreover, if $z^{(0)} \in B(z^*, \epsilon)$, we have that $\lim_{k\to\infty} z^{(k)} = z^*$ and by continuity it follows that $\lim_{k\to\infty} H_k = J_T(z^*)$.

ii) From the previous point we have that

$$z^{(k+1)} - z^* = (z^{(k)} - z^*) H_k \qquad \forall k \geq 0$$

with $H_k = -\operatorname{diag}(\Lambda^+(D_{z^*} - P^+)^{-1})(D_{z^{(k)}} - P^+)^{-1} P^- D_\mu^{-1}$. For all $k \geq 0$, we have that $z^{(k)} \geq \mathbf{1}$ and $z^* \geq \mathbf{1}$ (remark 4.2.5), so $(D_{z^*} - P^+)$ and $(D_{z^{(k)}} - P^+)$ are M-matrices (remark 4.2.2) and in particular $(D_{z^*} - P^+)^{-1}$ and $(D_{z^{(k)}} - P^+)^{-1}$ are nonnegative. Therefore $H_k \leq 0$ and the thesis follows.

iii) First of all, with the same reasoning as in remark 4.2.5, we have that $z_{\min} \leq z^{(k)}$ for all $k \geq 0$ and $z_{\min} \geq \mathbf{1}$ by the definition of $z_{\min}$ (remark 4.2.6). Recall now that the function T is non-increasing for $z \geq \mathbf{1}$ (see lemma 4.2.1), yielding that $T(z) \leq T(z_{\min}) = z^{(0)}$ for all $z \in \tilde{I}(z^*)$, and in particular $z^* \leq z^{(0)}$.

Using relation (4.2.21) we easily get the first part of the thesis:

$$z^{(2k)} \geq z^* \geq z^{(2k+1)} \quad \forall k \geq 0. \tag{4.2.24}$$

Now, observe that the composition $T \circ T$ is non-decreasing for $z \geq \mathbf{1}$. The proof is then by induction on k.

Since $T(z) \leq T(z_{\min}) \quad \forall z \in \tilde{I}(z^*)$, we have $z^{(2)} = T(z^{(1)}) \leq T(z_{\min}) = z^{(0)}$ and $z^{(3)} = T(z^{(2)}) \geq T(z^{(0)}) = z^{(1)}$, where we used that T is non-increasing and that $T \circ T$ is non-decreasing. Assuming now that $z^{(2k+1)} \leq z^{(2(k+1)+1)}$ and that $z^{(2k)} \geq z^{(2(k+1))}$ for a certain $k \geq 1$, applying $T \circ T$ yields

$$z^{(2(k+1)+1)} = (T \circ T)(z^{(2k+1)}) \leq (T \circ T)(z^{(2(k+1)+1)}) = z^{(2(k+2)+1)}$$

and

$$z^{(2(k+1))} = (T \circ T)(z^{(2k)}) \geq (T \circ T)(z^{(2(k+1))}) = z^{(2(k+2))},$$

proving the thesis.

Remark 4.2.7. The "alternating" property of lemma 4.2.4 *ii)* also holds for the sequence $(z^{(k+1)} - z^{(k)})_{k\geq 0}$, that is:

$$z^{(k)} - z^{(k-1)} \geq 0 \iff z^{(k+1)} - z^{(k)} \leq 0 \tag{4.2.25}$$

This property is useful also from a practical standpoint, since it allows to upper bound the error of each component at each step. In fact, the following bound holds:

$$|z_i^{(k)} - z_i^*| \leq |z_i^{(k)} - z_i^{(k-1)}| \qquad \forall i = 1, \dots, N$$

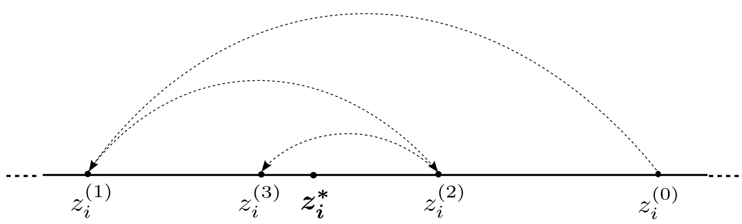

Figure 4.1: i-th component of the sequence $(z^{(k)})_{k\geq 0}$

□

The next theorem shows that the fixed point iteration (4.2.18) has a linear rate of convergence and the asymptotic reduction of the error is bounded by the spectral radius of the jacobian $J_T(z^*)$.

Theorem 4.2.3. *If $z^{(0)} \in \tilde{I}(z^*)$, where $\tilde{I}(z^*)$ is the neighbourhood defined in remark 4.2.6, the sequence* (4.2.18) *satisfies*

$$\lim_{k\to\infty} \sqrt[k]{\|z^{(k)} - z^*\|_\infty} \leq \rho(J_T(z^*)) \tag{4.2.26}$$

Proof. For $z \geq 1$, let $h(z) = -\mathrm{diag}(\Lambda^+(D_{z^*} - P^+)^{-1})(D_z - P^+)^{-1}P^- D_\mu^{-1}$, so that $H_k = h(z^{(k)})$ and $J_T(z^*) = h(z^*)$. Denote $J_* = J_T(z^*)$. The function h is component-wise non-decreasing. Applying h to the relation (4.2.22) yields

$$H_{2k-1} \leq H_{2k+1} \leq J_* \leq H_{2(k+1)} \leq H_{2k} \quad \forall k \geq 1 \tag{4.2.27}$$

Therefore in relation (4.2.20) we can upper bound the matrices H_k (distinguishing between odd and even indexes), obtaining

$$z^{(k+1)} - z^* = (z^{(0)} - z^*)H_0 \dots H_k = (z^{(0)} - z^*)H_0 \dots H_{2(l-1)}H_{2l-1}H_{2l} \dots H_k$$

where $0 \leq l < \lfloor k/2 \rfloor$ is a fixed nonnegative integer. Letting $w_l := (z^{(0)} - z^*)H_0 \dots H_{2(l-1)}$, and thanks to (4.2.27), we obtain

$$z^{(k+1)} - z^* \leq w_l \underbrace{(J_*H_{2l})(J_*H_{2l})\dots(J_*H_{2l})}_{\lfloor k/2 \rfloor - l + 1 \text{ times}} = w_l(J_*H_{2l})^{\lfloor k/2 \rfloor - l + 1}S$$

where

$$S = \begin{cases} I & \text{if } k \text{ is even} \\ J_* & \text{if } k \text{ is odd.} \end{cases}$$

We obtain

$$\|z^{(k+1)} - z^*\|_\infty \leq C_l \|(J_*H_{2l})^{\lfloor k/2 \rfloor + 1}\|_\infty$$

and

$$\sqrt[k]{\|z^{(k+1)} - z^*\|_\infty} \leq \sqrt[k]{C_l} \sqrt[k]{\|(J_*H_{2l})^{\lfloor k/2 \rfloor + 1}\|_\infty}$$

where

$$C_l = \|w_l\|_\infty \|S\|_\infty \|(J_*H_{2l})^{-l}\|_\infty.$$

Taking the limit for $k \to \infty$ yields

$$\lim_{k\to\infty} \sqrt[k]{\|z^{(k+1)} - z^*\|_\infty} \le \underbrace{\lim_{k\to\infty} \sqrt[k]{C_l}}_{=1} \lim_{k\to\infty} \sqrt[k]{\|(J_* H_{2l})^{\lfloor k/2 \rfloor + 1}\|_\infty} =$$
$$= \rho(J_* H_{2l})^{1/2}$$

where we used Gelfand's formula 1.2.1.

Taking now the limit for $l \to \infty$ yields

$$\lim_{k\to\infty} \sqrt[k]{\|z^{(k+1)} - z^*\|_\infty} \le \lim_{l\to\infty} \rho(J_* H_{2l})^{1/2} =$$
$$= \rho(J_*^2)^{1/2} = \rho(J_*)$$

where we used the continuity of the spectral radius and of the function h.

□

Complexity. The fixed point iteration (4.2.18) is an iterative method with linear rate of convergence, given by $\rho(J_T(z^*))$, as shown in Theorem 4.2.3. Each iteration requires the computation of the vector $T(z^{(k)}) = \Lambda^+(D_{z^{(k)}} - P^+)^{-1}P^- D_\mu^{-1} + \alpha D_\mu^{-1}$. The matrices P^+, P^-, D_μ^{-1} and the vectors Λ^+, α are given parameters. The only costly operation that we have to perform at each step is the solution of the linear system $v(D_{z^{(k)}} - P^+) = \Lambda^+$, since the other multiplications are just vector-matrix products, yielding a cost of $O(N^3)$ time per step.

4.2.3 Newton-Raphson iteration

In this section we will focus on the equation $S(z) := z - T(z) = 0$, since a zero of S is clearly a fixed point of T, and vice versa. Note that S is Fréchet differentiable with derivative $J_S(z) = I - J_T(z)$. As usual we will suppose that the system is ergodic.

We will consider the Newton-Raphson iteration:

$$z^{(k+1)} = z^{(k)} - S(z^{(k)}) J_S(z^{(k)})^{-1} = \tag{4.2.32a}$$
$$z^{(k)} - (z^{(k)} - T(z^{(k)}))(I - J_T(z^{(k)}))^{-1}, \qquad k \ge 0 \tag{4.2.32b}$$

The following result shows that, for a suitable choice of the starting vector z_0, the iteration (4.2.32) is well-defined and quadratically convergent to z^*.

Theorem 4.2.4. *Let z^* be the zero of function S. Then there exist $\delta, M > 0$ such that the iteration (4.2.32) is well-defined for all $z^{(0)} \in B(z^*, \delta)$ and*

$$\lim_{k\to\infty} \|z^{(k)} - z^*\| = 0$$

$$\|z^{(k+1)} - z^*\| \leq M\|z^{(k)} - z^*\|^2 \qquad \forall k \geq 0$$

Proof. Since $\rho(J_T(z^*)) < 1$ by theorem 4.2.1, $J_S(z^*) = I - J_T(z^*)$ is non-singular, and by lemma 4.2.3 J_S is also Lipschitz continuous for $z \geq \mathbf{1}$. Therefore there exist a $\delta_1 > 0$ such that $J_S(z)$ is Lipschitz continuous, non-singular and such that $\|J_S(z)^{-1}\| \leq K$ for all $z \in B(z^*, \delta_1)$, for a certain costant $K \in \mathbb{R}_+$.

By the mean value theorem 1.3.1 we have

$$S(z^{(k)}) = S(z^{(k)}) - S(z^*) = \int_0^1 (z^{(k)} - z^*)J_S(z^{(k)} + t(z^{(k)} - z^*))dt$$

Let $M = \frac{LK}{2}$, $0 < \delta < \min\{\delta_1, \frac{1}{M}\}$ and $z^{(k)} \in B(z^*, \delta)$. We have

$$\begin{aligned}
&z^{(k+1)} - z^* = z^{(k)} - S(z^{(k)})J_S(z^{(k)})^{-1} - z^* = \\
&\Big[(z^{(k)} - z^*)J_S(z^{(k)}) - S(z^{(k)})\Big]J_S(z^{(k)})^{-1} = \\
&\Big[(z^{(k)} - z^*)J_S(z^{(k)}) - \int_0^1 (z^{(k)} - z^*)J_S(z^{(k)} + t(z^{(k)} - z^*))dt\Big]J_S(z^{(k)})^{-1} = \\
&\Big[\int_0^1 (z^{(k)} - z^*)(J_S(z^{(k)}) - J_S(z^{(k)} + t(z^{(k)} - z^*)))dt\Big]J_S(z^{(k)})^{-1}.
\end{aligned}$$

Since $z^{(k)}, z^{(k)} + t(z^{(k)} - z^*) \in B(z^*, \delta)$ for $0 < t < 1$, we have

$$\|J_S(z^{(k)}) - J_S(z^{(k)} + t(z^{(k)} - z^*))\| \leq Lt\|z^{(k)} - z^*\|, \quad \|J_S(z^{(k)})^{-1}\| \leq K,$$

yielding

$$\begin{aligned}
&\|z^{(k+1)} - z^*\| \leq K\int_0^1 Lt\|z^{(k)} - z^*\|^2 = \frac{LK}{2}\|z^{(k)} - z^*\|^2 = M\|z^{(k)} - z^*\|^2, \\
&\|z^{(k+1)} - z^*\| < \underbrace{(M\delta)}_{<1}\|z^{(k)} - z^*\|
\end{aligned}$$

which is the thesis.

□

Complexity. Each step requires the computation of the vector $T(z^{(k)}) = \Lambda^+(D_{z^{(k)}} - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1}$ and of the matrix $(I - J_T(z^{(k)}))^{-1} = (I + \text{diag}(\Lambda^+(D_{z^{(k)}} - P^+)^{-1})(D_{z^{(k)}} - P^+)^{-1}P^-D_\mu^{-1})^{-1}$, yelding a $O(N^3)$ cost per step. In practice it is convenient to first compute the inverse $(D_{z^{(k)}} - P^+)^{-1}$. Then we obtain $T(z^{(k)}) = \Lambda^+(D_{z^{(k)}} - P^+)^{-1}P^-D_\mu^{-1} + \alpha D_\mu^{-1}$ and $I - J_T(z^{(k)}) = I + \text{diag}(\Lambda^+(D_{z^{(k)}} - P^+)^{-1})(D_{z^{(k)}} - P^+)^{-1}P^-D_\mu^{-1}$ with just vector-matrix products, matrix-matrix product with a diagonal factor and matrix sums.

A step of the Newton-Raphson method is therefore slightly more costly than a step of the Fixed Point iteration. However, the quadratic rate of convergence of Newton-Raphson makes it up for that, resulting in a much faster method.

Chapter 5

Numerical results

Let $(\overline{q}^{(k)})_{k\geq 0}, (z_{FP}^{(k)})_{k\geq 0}, (z_{NR}^{(k)})_{k\geq 0}$ be the Fourneau (4.1.2), Fixed Point (4.2.18) and Newton-Raphson (4.2.32) iterations, respectively. In the following they will be denoted as *FRN*,*FP*,*NR*, respectively. In this chapter numerical results regarding the convergence rates, the alternating properties and the CPU elapsed time will be presented. The software used is MatLab ver. 7.5.0. We consider three different structures of the matrices P^+ and P^-: a convex combination of each other, an almost triangular and a tridiagonal structure. The results show that when the dimension of the G-network is small or moderate, *NR* is faster than *FRN* and *FP*, making it a preferable choice in applications such the one described in chapter 3.

The starting point for the Fourneau iteration is given by (4.1.3), while for the Fixed Point and the Newton-Raphson we set $z^{(0)} = T(\mathbf{1})$ as an euristical choice (see also lemma 4.2.4). The relative errors for the three methods are denoted by

$$e_{FRN}^{(k)} = \frac{\|\overline{q}^{(k)} - \overline{q}^*\|}{\|\overline{q}^*\|}, e_{FP}^{(k)} = \frac{\|z_{FP}^{(k)} - z_{FP}^*\|}{\|z_{FP}^*\|}, e_{NR}^{(k)} = \frac{\|z_{NR}^{(k)} - z_{NR}^*\|}{\|z_{NR}^*\|} \tag{5.0.1}$$

for $k \geq 0$, where $\overline{q}^*, z_{FP}^*$ and z_{NR}^* are the solutions computed by the three methods.

We perform various experiments varying the structure of matices P^+, P^- and the magnitude of the vectors $\mu, \Lambda^+, \Lambda^-$.

5.1 Convex combination

We generate a stochastic matrix A with uniformly distributed psuedo-random elements in the interval $(0, 1)$ and, for a given $x \in (0, 1)$, we set

$$\begin{aligned} &D = \text{diag}(a_{11}, \ldots, a_{NN}) \\ &P^+ = x(A - D) \\ &P^- = (1 - x)(A - D) \end{aligned}$$

so that $P^+ + P^- + D = A$ is stochastic. In this way P^+ and P^- are both full matrices, and varying x we control the internal flow of customers in the network: the larger x, the larger the probability that a customers, after being served, leaves a queue as a positive customer rather than as a negative one.

Letting $\mathbf{1} = [1, \ldots, 1]$, we also set, for a given $k \in \mathbb{R}_+$,

$$\begin{aligned} &\mu = h\mathbf{1} \\ &\Lambda^+ = \mathbf{1} \\ &\Lambda^- = k\mathbf{1}. \end{aligned}$$

Varying k we can control the amount of negative customers arriving in the network from the outside, while varying h we control the service rate of the queues. The external arrival rate of positive customers Λ^+ is kept fixed.

The parameters x, h, k have been selected in such a way that the computed probabilities q_i are strictly less than one, ensuring the ergodicity of the network. In all the experiments all the three methods converged to the correct solution, confirming that $z^{(0)} = T(1)$ is a good euristical choice for the starting point. Below are riported the relative errors $e^{(k)}_{FRN}, e^{(k)}_{FRN}, e^{(k)}_{FRN}$ in typical scenarios. We used $N = 10$ in all the experiments.

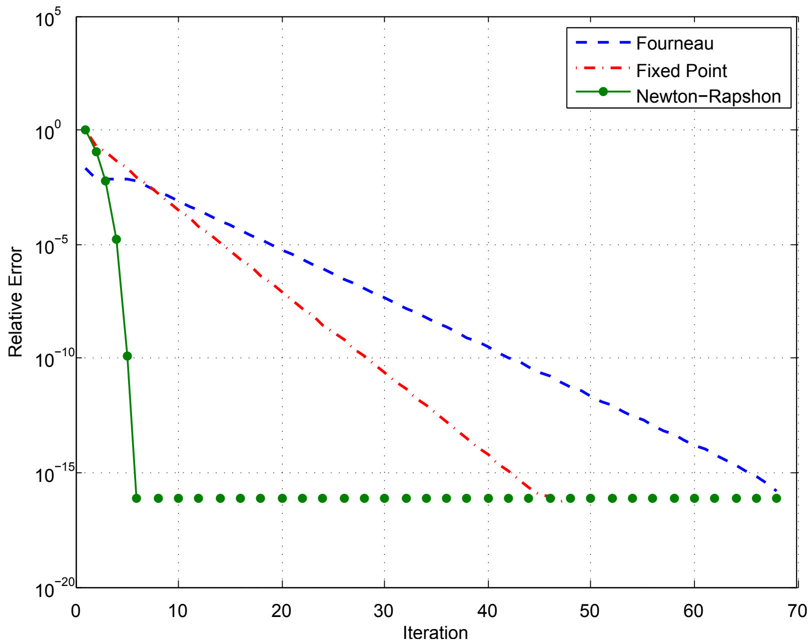

Figure 5.1: Convex combination structure. Relative errors for $x = 0.5$, $h = 1$, $k = 0.01$.

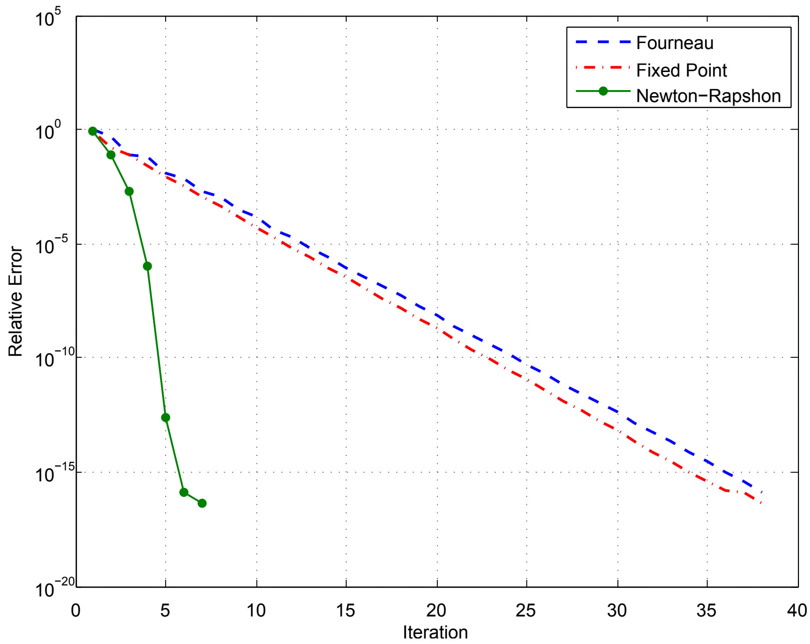

Figure 5.2: Convex combination structure. Relative error errors for $x = 0.05$, $h = 1$, $k = 0.01$.

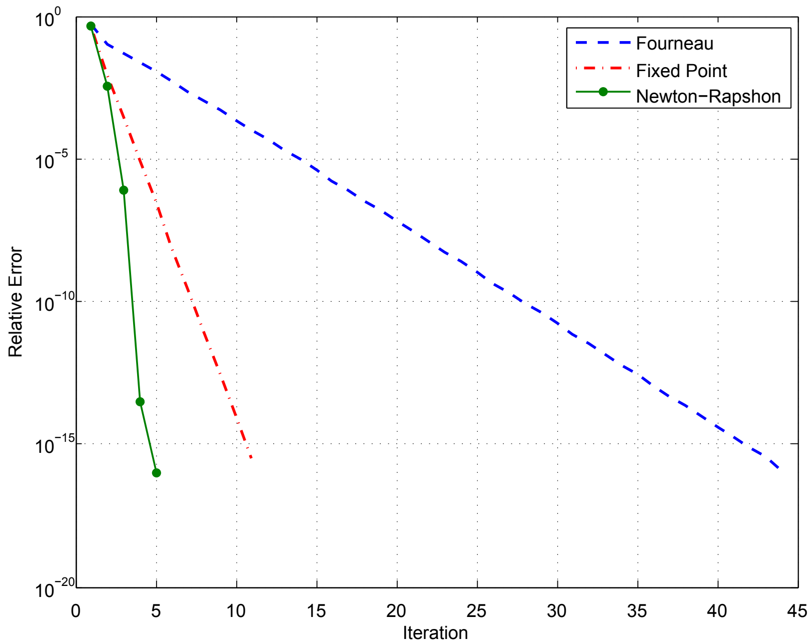

Figure 5.3: Convex combination structure. Relative error errors for $x = 0.95$, $h = 1$, $k = 1$.

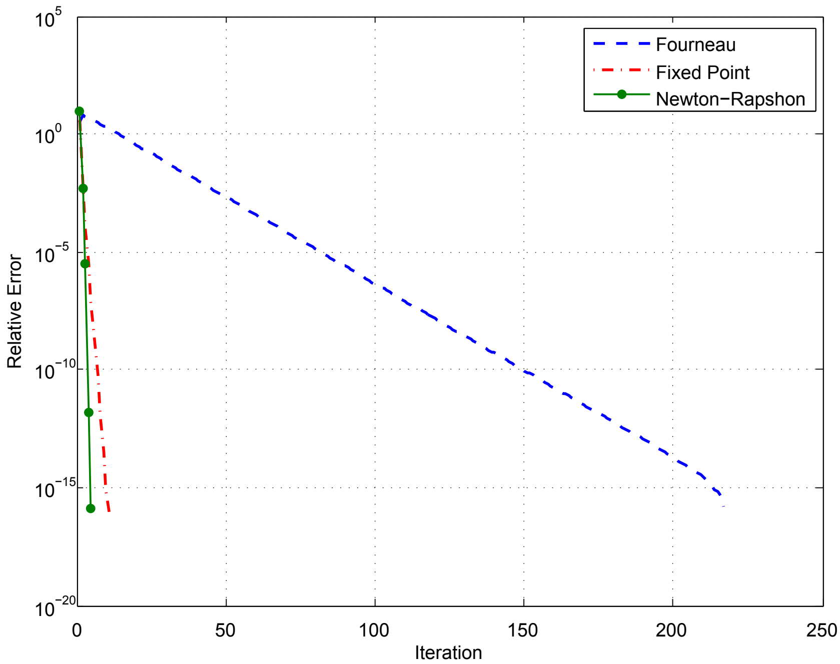

Figure 5.4: Convex combination structure. Relative errors for $x = 0.95$, $h = 100$, $k = 1$.

These plots confirm that *FRN* and *FP* have linear rate of convergence, while the Newton-Raphson iteration has quadratic rate.

In all cases *FP* converges faster than *FRN*, but the two methods perform differently: for small x (see fig. 5.2) *FRN* is particularly fast, reaching the same speed of *FP*, while for large x (see fig. 5.3) *FP* nearly reaches *NR* speed.

In all cases *NR* requires only a few iterations to reach convergence, outperforming *FP*, since both methods requires $O(N^3)$ time per step. In section 5.1.1 a performance comparison between *NR* and *FRN* will be carried out: since *FRN* requires only $O(N^2)$ time per step, there will be a threshold $\overline{N}$ such that *NR* converges faster than *FRN* for $N \leq \overline{N}$. *FRN* requires a particular large number of iterations when x is close to one and the service rate μ is large, as we can see in fig. 5.4.

Table 5.1 reports, for $x = 0.05, 0.5, 0.95$, $h = 1, 100$ and $k = 0.01, 1, 100$:

- the number of iterations necessary to reach a relative error of the order of 10^{-16}, for the three iterations;
- the maximum q_i, for $i = 1, \ldots, N$;

- the spectral radius $\rho(J_*) = \rho(J_T(z^*))$ of the Jacobian matrix of $T(z)$ in z^*;

- the convergence ratio of the Fourneau iteration

$$R_{FRN} = \lim_{k\to\infty} \frac{\|z^{(k+1)}_{FRN} - z^*_{FRN}\|}{\|z^{k}_{FRN} - z^*_{FRN}\|}$$

  for $k \geq 0$.

Remark 5.1.1. The convergence ratio of the fixed point iteration is not reported as in all cases it resulted equal to the upper bound $\rho(J_T(z^*))$ (see 4.2.3) up to four decimal figures.

In bold are the rows relative to the plots in fig. **??**− 5.4. We make the following remarks:

- as one would expect, the smaller k (i.e. the smaller the negative customers outside arrival rate) the higher the q_i-s (i.e. the higher the probability that in steady-state the queues are non-empty), and the smaller x (i.e. the smaller the positive customers traffic inside of the network) the smaller the q_i-s;

- $J_T(z^*)$ is decreasing in k;

- the number of iteration of *FRN* is increasing in x;

- the number of iterations of *FP* is decreasing in x;

- increasing the service rate ($h = 100$) reduces the occupation probabilities q_i-s, as one would expect, and it has opposite effects on the performance of *FNR*: for small x (i.e. the internal traffic is mostly constituted by *negative* customers), it reduces the number of iterations, while for large x, it increases it.

Table 5.1: Convex combination structure. Performance measures.

					$\mathbf{x = 0.05}$		
h	k	*FRN*	*FP*	*NR*	$\max q_i$	$\rho(J_*)$	R_{FRN}
1	$\mathbf{10^{-2}}$	**38**	**36**	**6**	$\mathbf{7.39 \cdot 10^{-1}}$	$\mathbf{3.58 \cdot 10^{-1}}$	$\mathbf{3.7 \cdot 10^{-1}}$
1	1	22	20	5	$4.49 \cdot 10^{-1}$	$1.58 \cdot 10^{-1}$	$1.73 \cdot 10^{-1}$
1	10^2	7	6	4	$1 \cdot 10^{-2}$	$8 \cdot 10^{-5}$	$5 \cdot 10^{-4}$
10^2	10^{-2}	15	9	4	$1.05 \cdot 10^{-2}$	$9.2 \cdot 10^{-3}$	$5.42 \cdot 10^{-2}$
10^2	1	15	9	4	$1.1 \cdot 10^{-2}$	$9 \cdot 10^{-3}$	$5.4 \cdot 10^{-2}$
10^2	10^2	13	6	3	$5 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
					$\mathbf{x = 0.5}$		
h	k	*FRN*	*FP*	*NR*	$\max q_i$	$\rho(J_*)$	R_{FRN}
1	$\mathbf{10^{-2}}$	**68**	**45**	**6**	$\mathbf{9.94 \cdot 10^{-1}}$	$\mathbf{4.39 \cdot 10^{-1}}$	$\mathbf{6.11 \cdot 10^{-1}}$
1	**1**	**30**	**19**	**5**	$\mathbf{5.65 \cdot 10^{-1}}$	$\mathbf{1.39 \cdot 10^{-1}}$	$\mathbf{3.11 \cdot 10^{-1}}$
1	10^2	9	5	4	$1 \cdot 10^{-2}$	$4 \cdot 10^{-5}$	$4 \cdot 10^{-3}$
10^2	10^{-2}	52	11	5	$1.94 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	$4.46 \cdot 10^{-1}$
10^2	1	50	10	5	$1.91 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$4.44 \cdot 10^{-1}$
10^2	10^2	28	6	3	$7 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2.18 \cdot 10^{-1}$
					$\mathbf{x = 0.95}$		
h	k	*FRN*	*FP*	*NR*	$\max q_i$	$\rho(J_*)$	R_{FRN}
1	**1**	**46**	**12**	**5**	$\mathbf{9.35 \cdot 10^{-1}}$	$\mathbf{3.17 \cdot 10^{-2}}$	$\mathbf{4.53 \cdot 10^{-1}}$
1	10^2	12	5	4	$1 \cdot 10^{-2}$	$6 \cdot 10^{-6}$	$9 \cdot 10^{-3}$
10^2	10^{-2}	207	10	5	$7.22 \cdot 10^{-2}$	$1.65 \cdot 10^{-2}$	$8.38 \cdot 10^{-1}$
$\mathbf{10^2}$	**1**	**175**	**9**	**5**	$\mathbf{6.4 \cdot 10^{-2}}$	$\mathbf{1.12 \cdot 10^{-2}}$	$\mathbf{8.06 \cdot 10^{-1}}$
10^2	10^2	50	5	3	$1 \cdot 10^{-2}$	$4 \cdot 10^{-4}$	$4.36 \cdot 10^{-1}$

We conclude that:

- the smaller the value of Λ^-, the slower the convergence of *FP* and *FRN*;
- *FRN* performs better for small x, while *FP* performs better for large x. In any case, *FP* requires less iterations than *FRN*;
- *FRN* performs particularly well for small x and large μ, and particularly bad for large x and large μ;
- *NR* requires less iterations than *FRN* and *FP* in all cases.

Next we analyze the monotonicity properties of the three methods. In fig. 5.5 and fig. 5.6 are reported the first component of the FRN iteration, namely the $\overline{q}_1{}^{(k)}$, and the first components of the FP iteration, namely the $(z_1)^{(k)}_{FP}$, together with the corresponding values for the occupation probabilities, i.e. $(q_1)^{(k)}_{FP}$. Recall that the relation between z and q is

$$q = \Lambda^+(D^*_z - P^+)^{-1}D^{-1}_\mu$$

(see remark 4.2.3).

We can see that the FRN iteration is monotonically decreasing to $\overline{q}_1{}^*$, as proven by Fourneau in [4], while the FP iteration $z^{(k)}_{FP}$ alternates around the fixed point z^*, as proven in lemma 4.2.4. Note that also $(q_1)^{(k)}_{FP}$ alternates, simmetically with respect to $z^{(k)}_{FP}$, since the function $z \to \Lambda^+(D^*_z - P^+)^{-1}D^{-1}_\mu$ is non-increasing in $z \geq \mathbf{1}$.

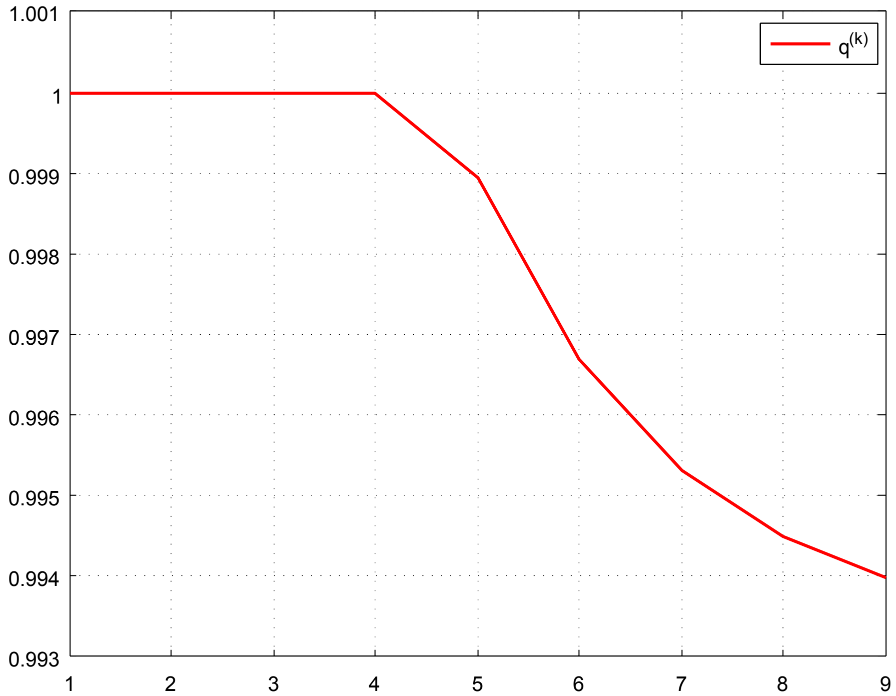

Figure 5.5: Convex combination structure. First component of the Fourneau iteration for $x = 0.5$, $h = 1$, $k = 0.01$.

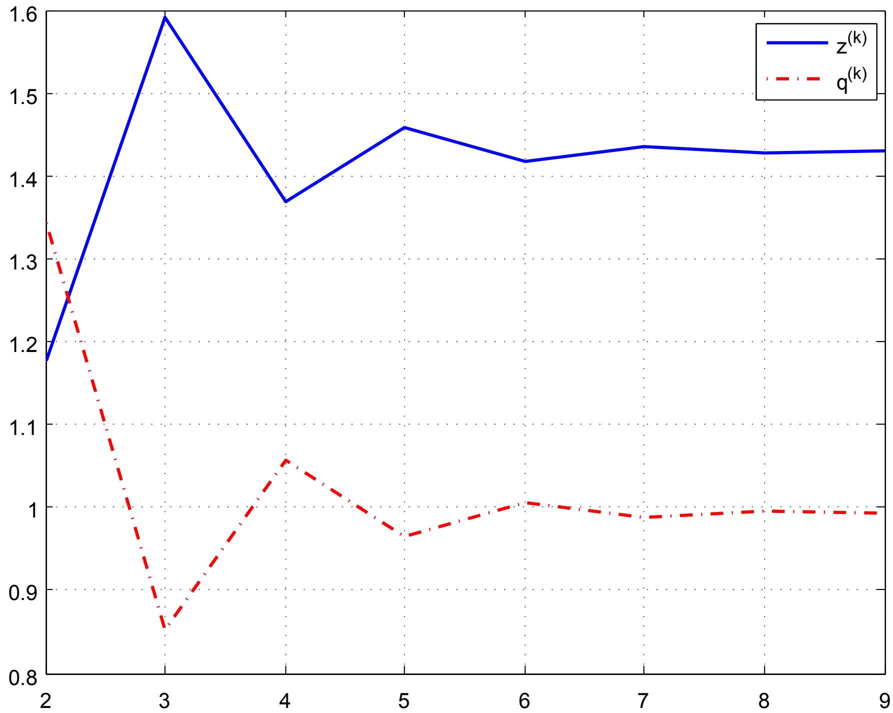

Figure 5.6: Convex combination structure. First component of the Fixed Point iteration for $x = 0.5$, $h = 1$, $k = 0.01$.

5.1.1 CPU time

In this section we will compare the CPU time requested by the Fourneau and the Newton-Raphson iteration for several value of the dimension N, using the convex combination structure for the matrices P^+ and P^-. We have shown that *NR* requires just a few iterations to reach convergence, while *FRN*, in some cases, can be very slow. This is due to the fact that *NR* has a quadratic rate of convergence, while *FRN* has a linear rate which can be close to 1 in some cases. However, *FRN* has a quadratic cost per step, while *NR* has a cubic cost per step: therefore we expect that for each case there will be a *threshold* $\overline{N}$ such that *NR* is preferable only for the dimensions $N \leq \overline{N}$. This property will make *NR* a suitable algorithm for the Traffic Matrix estimation problem, described in 3.2.

Remark 5.1.2. We do not perform the same comparison between *FRN* and *FP* as they are both linearly convergent methods, while *FP* has a greater cost per step than *FRN*, making it slower in almost all cases.

Below we report the results for the following case studies, where *FRN* performs differently.

- Case A: $x = 0.05$, $\mu = 10^2 \cdot \mathbf{1}$,$\Lambda^+ = \Lambda^- = \mathbf{1}$ (FRN perform particularly good).
- Case B: $x = 0.5$, $\mu = 1 \cdot \mathbf{1}$,$\Lambda^+ = \mathbf{1}$, $\Lambda^- = 10^{-2} \cdot \mathbf{1}$ (FRN performance is average).
- Case C: $x = 0.95$, $\mu = 10^2 \cdot \mathbf{1}$,$\Lambda^+ = \Lambda^- = \mathbf{1}$ (FRN perform particularly bad).

For each case and for each value of N, several simulations have been performed. In the plots are reported the average CPU times over all the simulations.

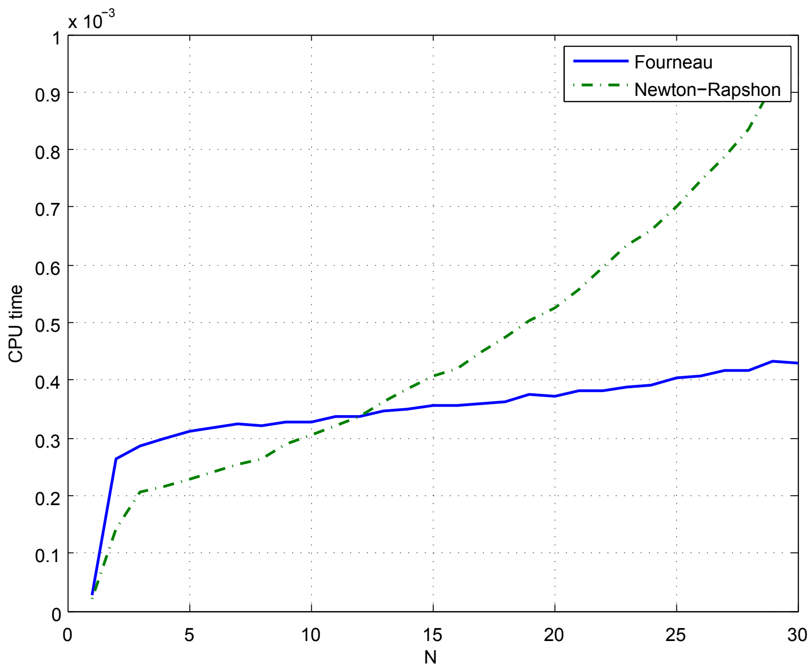

Figure 5.7: Convex combination structure. CPU times, case A.

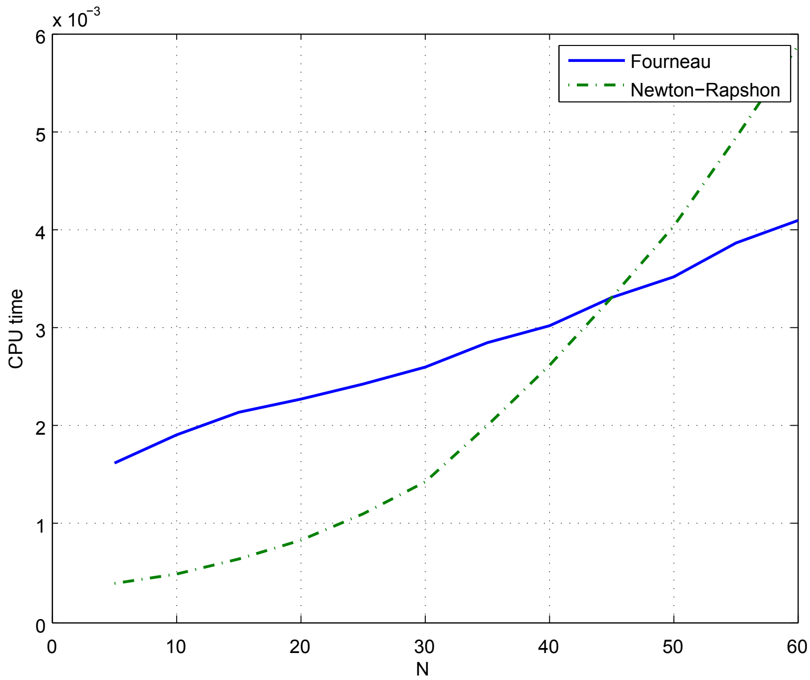

Figure 5.8: Convex combination structure. CPU times, case B.

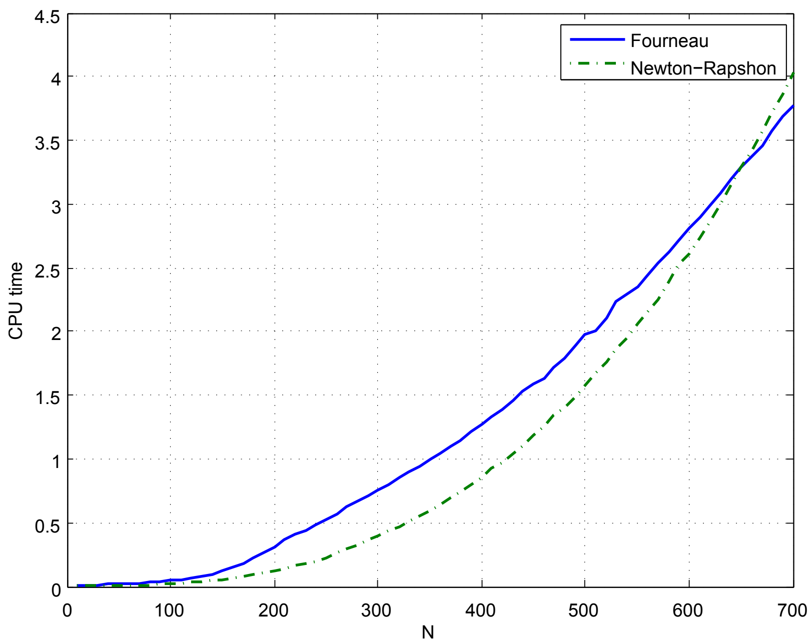

Figure 5.9: Convex combination structure. CPU times, case C.

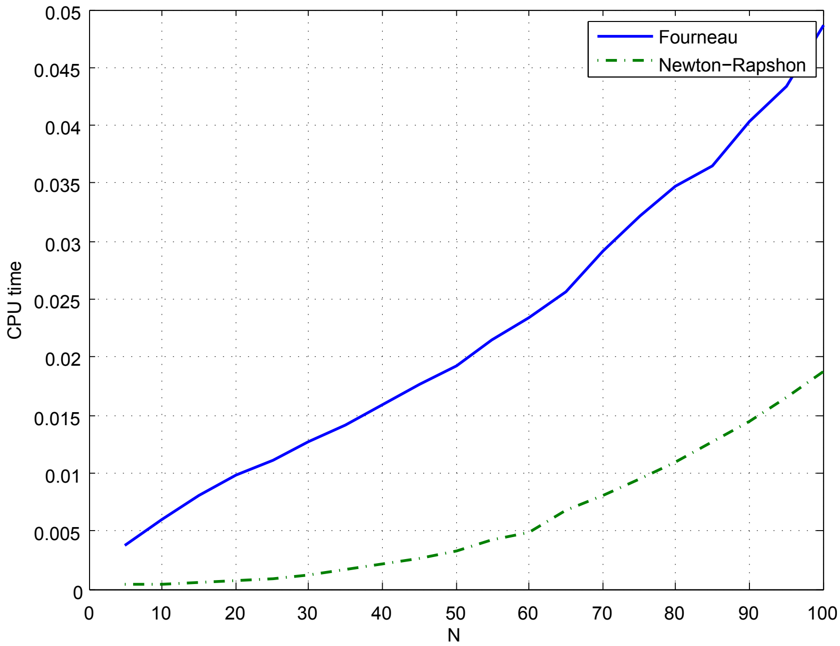

Figure 5.10: Convex combination structure. CPU times, case C. Zoom 1.

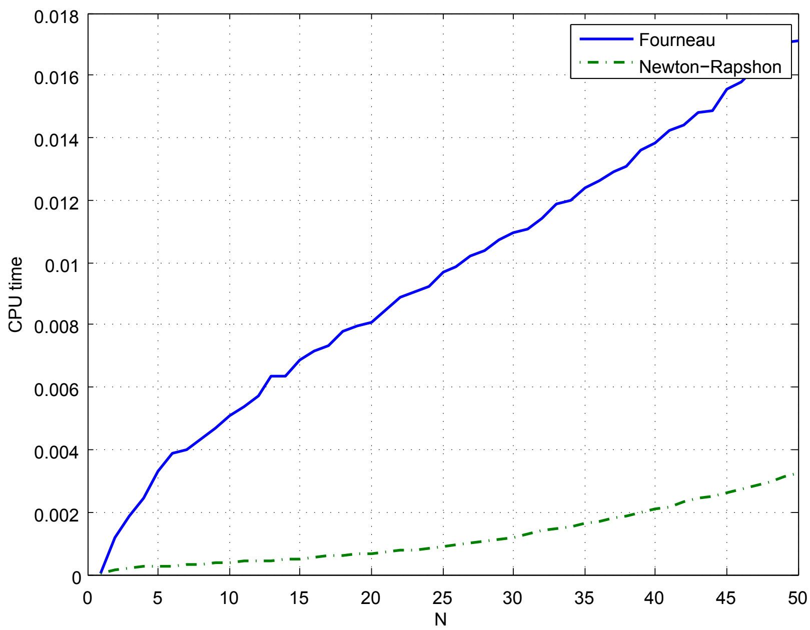

Figure 5.11: Convex combination structure. CPU times, case C. Zoom 2.

We observe that:

- In case A (fig. 5.7), the threshold is $\overline{N} = 14$: there is no practical reason to prefer *NR* over *FRN* in this case.
- In case B (fig. 5.8), the threshold is $\overline{N} = 44$: *NR* is faster than *FRN* for small values of N. For $N = 10$, the dimension used in the eperiments in table 5.1, *NR* is around 4 times faster than *FRN*.
- In case C (fig. 5.9), the threshold is around $\overline{N} = 620$: *FRN* is very slow in this case, making *NR* a preferable choice for moderate/large values of N. In particular, as reported in fig. 5.10,5.11, *NR* is around 13 times faster than *FRN* for $N = 10$, 7.5 times faster for $N = 40$, 3.5 times faster for $N = 70$ and roughly 2 times faster for larger values of N up to 300.

This results encourage the use of *NR* in problems where the the steady-state distributions of a large number of G-networks have to be computed, as in the context of supervised learning. In section 3.1) we described a learning algorithm and we observed that step $2a$) of that algorithm requires the solution of the system 3.1.4, i.e. the computation of the steady-state distribution: in [16] it is advised to use the Fourneau algorithm, due to its $O(N^2)$ cost per step. However, as we have shown in this section, there are cases in which the Newton-Raphson method is several times faster than the Fourneau iteration, at least up to a certain dimension $\overline{N}$.

Therefore, employing the Newton-Raphson method in those cases can improve the performance of the learning algorithm, as step 2) must be usually carried out a large number of times, proportional to the dimension of the learning dataset.

In particular, the Traffic Matrix estimation problem (section 3.2) is a scenario in which using the Newton-Raphson method could be really beneficial, since the G-network dimensions are usually small. In [14] the authors takled the problem by means of training several G-networks, each having a particular simple three-layer feed-forward structure and a small dimension, ranging from 9 to 14. They trained a total of $m = 132$ G-network using a learning dataset composet of 288 input-output pairs, and according to the gradient descent methods described in 3.1 this implies that the system (2.3.2) must be solved a number of times proportional to $132 \cdot 288 = 38016$.

In that particular case, due to the feed-forward structure, no particular algorithm is required for the solution of the correspondent systems (2.3.1), as described in section 2.5.1.

Nonetheless, the results developed in [14] suggest that:

- the approach can be refined by employing more complex G-networks, without restricting only to feed-forward structure. This includes:
  - matrices P^+, P^- having general structure.
  - external arrivals of positive customers occurring for all queues.
  - external arrivals of negative customers occurring for all queues.
- Given the dimension of m and of the datasets in real examples, the system (2.3.1) must be solved a large number of times. In [14] this number is of the order of 10^4.
- There is a need for efficient algorithms for the solution of the system (2.3.1).
- The G-network dimension can be reasonably supposed small/moderate, given the encouraging results already obtained with dimensions $N \leq 14$.

Given the results regarding CPU times presented in this section, it is clear that employing the Newton-Raphson algorithm in this setting can substantially improve the performances.

5.2 Almost triangular

We generate a stochastic matrix A with uniformly distributed psuedo-random elements in the interval $(0, 1)$. We then set $D = \mathrm{diag}(a_{11}, \ldots, a_{NN})$ and let P^+, P^- be equal, respectively, to the strictly lower and strictly upper triangular part of the matrix A, plus a correction for $p^+_{1,n}$ and $p^-_{n,1}$ in order to obtain irreducible matrices. Therefore P^+, P^- have the form

$$P^+ = \begin{pmatrix} 0 & 0 & 0 & 0 & \times \\ \times & 0 & 0 & 0 & 0 \\ \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 \end{pmatrix}, P^- = \begin{pmatrix} 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times \\ \times & 0 & 0 & 0 & 0 \end{pmatrix}$$

This is a variant of the simpler feed-forward networks (2.5.1), where both matrices P^+ and P^- are strictly upper triangular and a direct solution method exists.

We also set, for a given $k \in \mathbb{Z}$,

$$\begin{aligned} \mu &= \mathbf{1} \\ \Lambda^+ &= \Big[\frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}, \frac{N}{N}\Big] \\ \Lambda^- &= 10^k \Big[\frac{N}{N}, \frac{N-1}{N}, \ldots, \frac{2}{N}, \frac{1}{N}\Big]. \end{aligned}$$

Λ^+, Λ^- are chosen in this way in order to balance the flow of positive and negative customers to each queue. For example the first queue receives positive customers by all the other queues, so the external rate of positive customers at the first queue is reduced consequently. Conversely, the last queue receive no positive customers from any of the other queues, so its external positive arrival rate is larger. The situation is symmetrical for the negative customers.

Fig 5.12-5.13 report the plot of the relative errors for the cases with $k = 0$ and $k = 2$, when $N = 10$. The same observations as in section 5.1 hold.

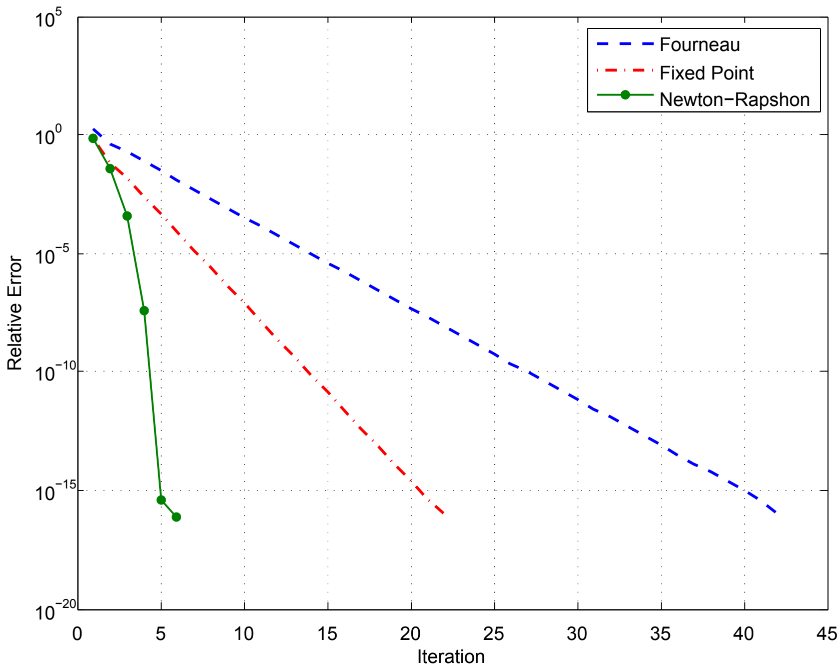

Figure 5.12: Almost triangular structure. Relative errors for $k = 0$.

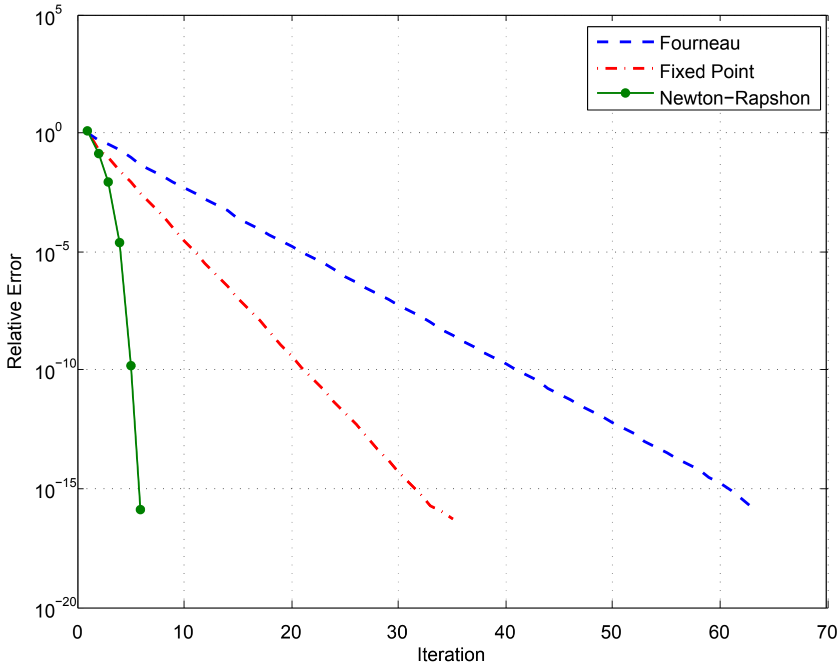

Figure 5.13: Almost triangular structure. Relative errors for $k = 2$.

5.3 Tridiagonal

For $N = 10$, we generate a stochastic matrix A with uniformly distributed psuedo-random numbers in the interval $(0, 1)$ for the off-diagonal entries, while for the diagonal entries the valules are generated in the interval $(0, y)$, for $y \in \mathbb{R}$. Varying y we control the probability that a customer exits the network, i.e. the magnitude of the matrix D.

We then set:

- $D = \text{diag}(a_{11}, \ldots, a_{NN})$
- P^- is the tridiagonal matrix with null principal diagonal and having the 1-st and -1-st diagonals of A as 1-st and -1-st diagonals.
- $P^+ = A - D - P^-$

Therefore P^+, P^- have the form

$$P^- = \begin{pmatrix} 0 & \times & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 \\ 0 & \times & 0 & \times & 0 \\ 0 & 0 & \times & 0 & \times \\ 0 & 0 & 0 & \times & 0 \end{pmatrix} P^+ = \begin{pmatrix} 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ \times & 0 & 0 & 0 & \times \\ \times & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \end{pmatrix}$$

In fig. 5.14-5.15 we report the relative errors $e^{(k)}_{FRN}, e^{(k)}_{FP}, e^{(k)}_{NR}$ in three particular cases. We set $\mu = 10^2 \cdot \mathbf{1}, \Lambda^+ = \Lambda^- = \mathbf{1}$ for both cases, while we set $y = 10^3, 0, 10^{-3}$.

We observe that with these parameters, the Fourneau iteration performs well only for large values of y (fig. 5.14). For smaller values of y (5.15-5.16) *FP* and *NR* outperform *FRN* by a large amount.

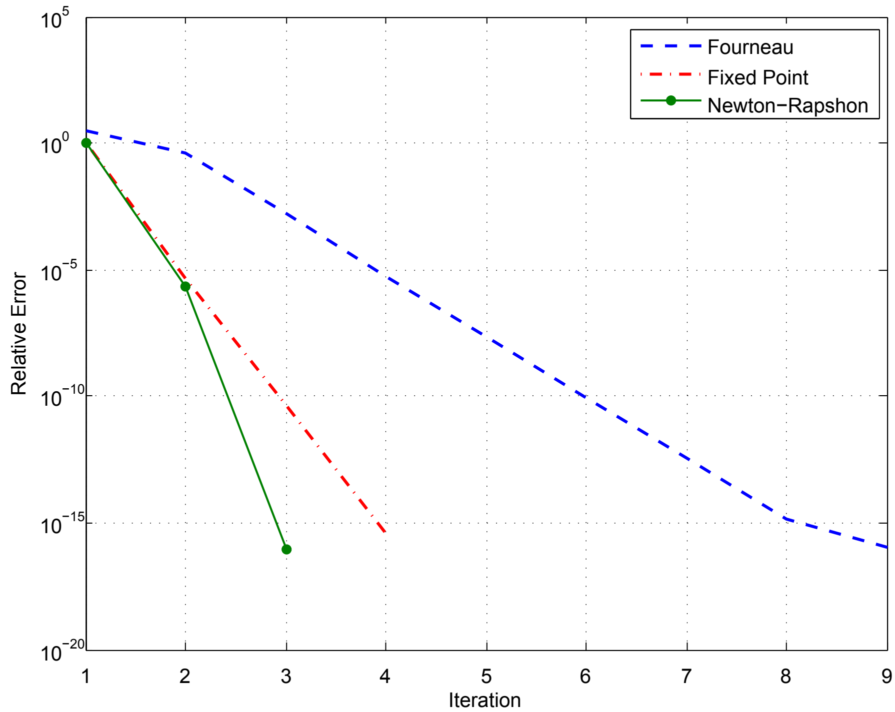

Figure 5.14: Tridiagonal structure. Relative errors for $y = 10^3$.

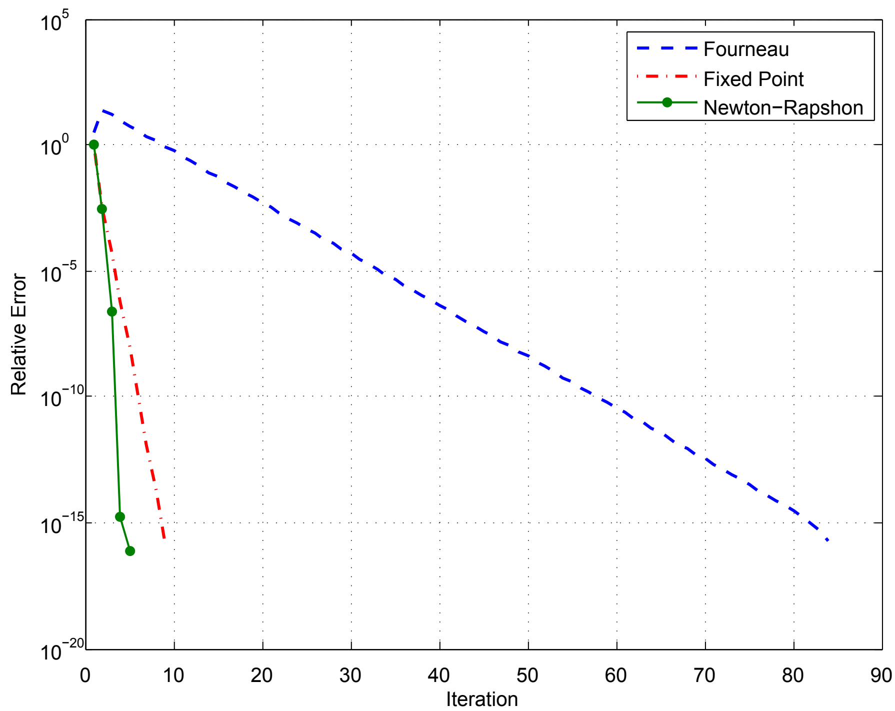

Figure 5.15: Tridiagonal structure. Relative errors for $y = 10^1$.

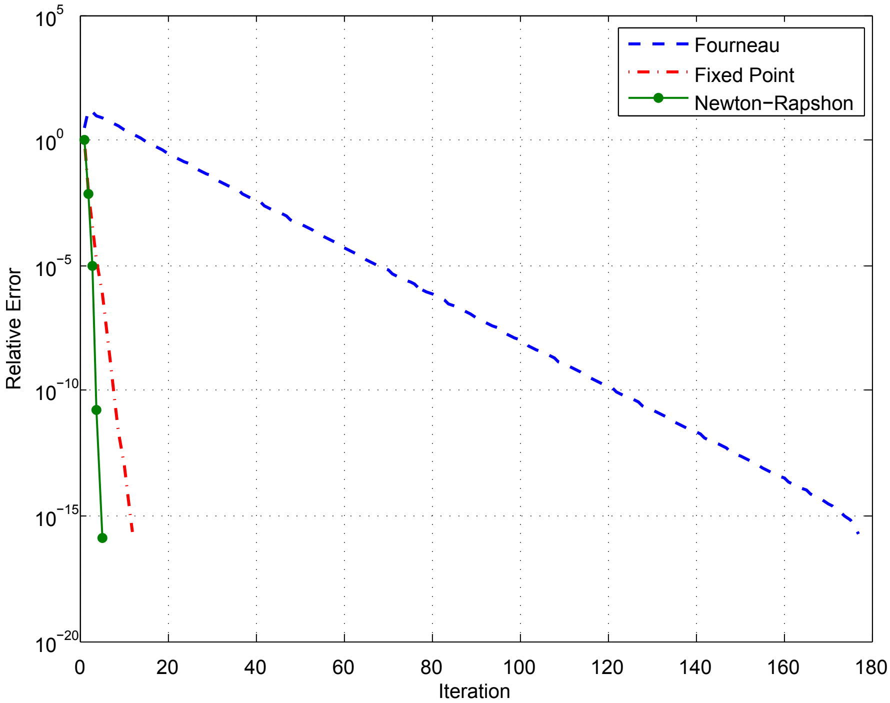

Figure 5.16: Tridiagonal structure. Relative errors for $y = 10^{-3}$.

Conclusion and future work

In this thesis we have studied a particular class of queueing networks called *G-networks*, also known as *Random neural networks.* The main characteristic of G-networks is the presence of *negative customers*, in addition to the usual customers in a queueing network. We have reviewed the main theoretical properties of G-networks, focusing on the existence of the stationary distribution of the number of customers in the network. In particular, we saw that, under stability condition, the steady-state distibution exists and it is given in product form, which is a desiderable trait.

However, finding the steady-state distribution involves the solution of a nonlinear system of equation ((2.3.1)), which is a challenging numerical problem and the main scope of this thesis. We have presented an equivalent formulation of the system (2.3.1) in terms of a matrix fixed point equation, which has allowed us to develop and study two new numerical methods, namely a fixed point iteration and a Newton-Raphson iteration.

We have proved that the fixed point iteration is locally convergent with a linear rate of convergent, that the iteration alternate around the fixed point and we provided a strict upper bound for the asymptotic reduction of the error. We then proved the well-posedness and the local convergence of the Newton-Raphons methods, showing that the convergence is quadratic.

We have then compared the performances of the two new methods with an existing algorithm, concluding that the Newton-Raphson is preferable for G-networks having small dimension and, in some cases, also for larger dimension.

Finally, we have suggested a possibile development of the results developed in this thesis in the context of the Traffic Matrix estimation problem, which has also been a motivating application in the first place. Refining the approach used in [14], we propose to consider fully connected G-networks and to employ the Newton-Raphson method in the learning algorithm. Due to the low dimensionality of the G-networks involved, and considered that the system (2.3.1) have to be solved a large number of times, we believe that the Newton-Raphson method would be a preferable choice with respect to existing methods.

Bibliography

[1] E. Gelenbe, “Random neural network with negative and positive signals and product form solution”, *Neural Computation*, Vol. 1, No. 4, pp. $502-510$, 1989.

[2] E. Gelenbe, “Stability of the random neural network model, *Proc. Of Neural Computation Workshop*, ISBN 3540522557, pp. $56-58$, Berlin, Germany, 1990.

[3] E. Gelenbe “Queueing networks with negative and positive customers”, *Journal of Applied Probability*, Vol. 28, No. 3, pp. $656-663$, 1991.

[4] J.M. Fourneau, “Computing the steady-state distribution of networks with positive and negative customers“, 13*th IMACS World Congress on Computation and Applied Mathematics*, Dublin, 1991.

[5] E. Gelenbe, M. Schassberger, “Stability of product form G-Networks”, *Probability in the Engineering and Informational Sciences*, Vol. 6, pp $271-276$, 1992.

[6] E. Gelenbe, “Learning in the recurrent random neural network”, *Neural Computation*, Vol. 5, No. 1, pp. $154-164$, 1993.

[7] E. Gelenbe, “G-networks with instantaneous customer movement”, *Journal of Applied Probability*, Vol. 30, No. 3, pp. $742-748$, 1993.

[8] E. Gelenbe, “G-Networks with signals and batch removal”, *Probability in the Engineering and Informational Sciences*, Vol. 7, pp. $335-342$, 1993.

[9] E. Gelenbe “G-networks: An unifying model for queuing networks and neural networks”, *Annals of Operations Research*, Vol. 48 $(1-4)$, pp. $433-461$, 1994.

[10] J.M. Fourneau, E. Gelenbe, R. Suros, “G-networks with multiple classes of positive and negative customers”, *Theoretical Computer Science*, Vol. 155, pp. $141-156$, 1996.

[11] J.R. Artalejo, “G-networks: a versatile approach for work removal in queueing networks”, *European J. Op. Res.*, Vol. 126, pp. 233 – 249, 2000.

[12] S Mohamed, G. Rubino, M. Varela, “Performance evaluation of real-time speech through a packet network: a random neural network-based approach”, *Performance Evaluation*, Vol. 57, No. 2, 141 – 161, 2004.

[13] G. Rubino, P. Tirilly, M Varela, “Evaluating Users’ Satisfaction in Packet Networks using random neural networks”, S.D. Kollias, A. Stafylopatis, W. Dutch, E. Oja (eds.), *ICANN* 2006, Vol. 4121, pp. 303 – 312, Springer, Heidelberg, 2006.

[14] P. Casas, S. Vaton, “On the Use of Random Neural Networks for Traffic Matrix Estimation in Large-Scale IP Networks”, *IWCMC '10 Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, pp. 326 – 330, 2010.

[15] E.Gelenbe, F. Batty, “Minimum graph covering with the random neural network model”, *Proc. ORSA Computer Science Technical Committee Conf.*, Williamsburg, VA, PergamonPress, 1992.

[16] S. Timotheou, “The Random Neural Network: a Survey”, *The Computer Journal*, Vol. 53 No. 3, pp. 251 – 267, 2010.

[17] V.Atalay, E.Gelenbe, N.Yalabik, “Image texture generation with the random neural network model, *Int. Conf. on Artificial Neural Networks (ICANN-91)*, Helsinki, June 1991.

[18] A. Berman, R.J. Plemmons, ”Nonnegative matrices in the mathematical sciences“, *Classics in Applied Mathematics*, 1979.

[19] J. M. Ortega, W. C. Rheinboldt, ”Iterative Solution of Nonlinear Equations in Several Variables“, *Classics in Applied Mathematics*, 1970.

[20] J. Walrand, P. Varaiya, ”Sojourn Times and the Overtaking Condition in Jacksonian Networks“, *Advances in Applied Probability*, Vol. 12, No. 4, pp. 1000-1018, 1980.

[21] F. Baskett, K. Many Chandy, R.R. Muntz, F.G. Palacios, "Open, closed and mixed networks of queues with different classes of customers", *Journal of the ACM*, Vol. 22, No. 2, pp. 248-260, 1975.

[22] R.S. Varga, ”Matrix Iterative Methods“, *Spring Series in Computational Mathematics*, Vol. 27, 1999.