

*MathematicS*  
*MathS in A.*  
*In Action*

BRANDA P. I. GONCALVES

**An Interacting Neuronal Network with Inhibition: theoretical analysis and perfect simulation**

Volume 12 (2023), p. 3-22.

<https://doi.org/10.5802/msia.29>

© Les auteurs, 2023.



Cet article est mis à disposition selon les termes de la licence CREATIVE COMMONS ATTRIBUTION 4.0.

<http://creativecommons.org/licenses/by/4.0/>



MathematicS In Action est membre du  
Centre Mersenne pour l'édition scientifique ouverte

<http://www.centre-mersenne.org/>

e-ISSN : 2102-5754

# An Interacting Neuronal Network with Inhibition: theoretical analysis and perfect simulation

BRANDA P. I. GONCALVES \*

\* Laboratoire de Physique Théorique et Modélisation, CY Cergy Paris Université, CNRS  
UMR-8089, 2 avenue Adolphe Chauvin, 95302 Cergy-Pontoise, France  
*E-mail address:* b.goncalves@parisnanterre.fr.

## Abstract

We study a purely inhibitory neural network model where neurons are represented by their state of inhibition. The study we present here is partially based on the work of Cottrell [6] and Fricker et al. [8]. The spiking rate of a neuron depends only on its state of inhibition. When a neuron spikes, its state is replaced by a random new state, independently of anything else, and the inhibition states of the other neurons increase by a positive value. Using the Perron–Frobenius theorem, we show the existence of a Lyapunov function for the process. Furthermore, we prove a local Doeblin condition which implies the existence of an invariant probability measure for the process. Finally, we extend our model to the case where the neurons are indexed by  $\mathbb{Z}$ . We construct a perfect simulation algorithm to show the recurrence of the process under certain conditions. To do this, we rely on the classical contour technique used in the study of contact processes, and assuming that the spiking rate takes values in the interval  $[\beta_*, \beta^*]$ , we show that there is a critical threshold for the ratio  $\delta = \frac{\beta_*}{\beta^* - \beta_*}$  over which the process is ergodic.

## 1. Introduction

For the operation of a neural network, neurons excite and/or inhibit each other. Here, we study a model of a purely inhibitory neural network where neurons are represented by their inhibitory state. The study we present is partially based on the work of Cottrell [6]. Her model consists of considering  $N$  interacting neurons described their state of inhibition. In her work, a neuron spikes when its state touches the value 0. When a neuron spikes, the state of inhibition of the other neurons increase by a non-negative deterministic constant  $\theta$ . The spiking neuron immediately receives a random inhibition independently of anything else. In Cottrell’s work the state of inhibition is just the waiting time until the next spike.

In the present work we generalize Cottrell’s model in several natural ways. Actually, in Cottrell’s model, the next spiking time in the neural net is deterministic and we will lift this assumption. A random spiking time is more realistic than a deterministic one since stochasticity is present all over in the brain functioning. Secondly, to allow formal general models we allow the state of inhibition to decrease at a general rate in between the successive spikes of the network while in Cottrell’s work the drift of the flow is equal to  $-1$ .

In the first part of this paper, we consider systems of  $N$  interacting neurons, in which any neuron can spike at any time. The spiking neuron takes a new random state of inhibition, and the others increase their inhibitory state by a deterministic quantity that we will call the inhibition weight, which depends on the distance between the spiking neuron and the “receiving” neuron, so that a neuron located far away of the spiking neuron is less impacted by the spike. The model

---

This research was conducted within the part of the Labex MME-DII(ANR11-LBX-0023-01) project and the CY Initiative of Excellence (grant “Investissements d’Avenir” ANR-16-IDEX-0008), Project EcoDep PSI-AAP 202-00000013.

*Keywords:* Spiking rate, Interacting neurons, Perfect simulation algorithm, Classical contour technique.  
*2020 Mathematics Subject Classification:* 00X99.

thus presented obviously extends Cottrell [6] and Fricker et al. [8] in two ways: the spiking time is no more deterministic but it is random; the dynamic of the process is no more constant.

Firstly, we show the existence of a Lyapunov function that allows us to formulate a sufficient condition of non-evanescence of the process in the sense of Meyn and Tweedie [14], i.e. a condition ensuring that the process does not escape at infinity. To do so, we introduce a reproduction matrix  $H$  and we suppose the spectral radius of  $H$  is lower than 1. The eigenvector associated with the spectral radius of  $H$  allows us to find a Lyapunov function for the process.

Secondly, we study the recurrence of the process relying on Doeblin conditions which we establish for the embedded chain sampled at the jump times. We show the existence of an invariant probability measure for the process. We do this in the case when the distribution of the new states has an absolutely continuous density and the jump rate is bounded.

In a second part, we consider the case where we have an infinite number of neurons indexed by  $\mathbb{Z}$  (see Comets et al. [4], Galves and Löcherbach [10], Galves et al. [11] and Morgan André [1]). The mean field behavior of such models has been studied by Cormier et al. [5] and Robert and Touboul [15] who were also interested in the stationary distributions of these processes. In the work of Ferrari et al. [7], considering an infinite system of interacting point processes with memory of variable length, the authors investigated the conditions for the existence of a phase transition using the classical contour technique, based on the classical work of Griffeath [13] on a contact process. Morgan André in this work [1], proves that the model described in [7] presents a metastable behavior while relying on the contour technique used in [13]. Following the idea of Ferrari et al. [7], Galves et al. [9] and Griffeath [13], we construct a perfect simulation algorithm that allows us to show the recurrence of the process. Assuming that the spiking rate takes values in the interval  $[\beta_*, \beta^*]$ , we show that there is a critical threshold for the ratio  $\delta = \frac{\beta_*}{\beta^* - \beta_*}$  over which the process is ergodic.

This paper is organized as follows. In Section 2 we describe the model and study the law of the first jump time of the process. The Foster–Lyapunov and Doeblin conditions are discussed to find non-evanescence criteria and to show the existence of a unique invariant probability measure of the process in Section 3 which is our first main result. Finally, in Section 4, we present a perfect simulation algorithm and we simulate the law of the state of inhibition of a given neuron in its invariant regime.

## 2. The model

### 2.1. Description of the model

In our paper, let us consider we have  $N$  neurons that are related to each other. For all  $i \in \{1, \dots, N\}$ ,  $X_t^{i,N}$  describes the state of inhibition of neuron  $i$  at time  $t$ . When the neuron  $i \in \{1, \dots, N\}$  spikes,

- The current state of inhibition of neuron  $i$  is replaced by a new value  $Y^i$  independently of anything else with distribution  $F^i$ .  $Y^i$  is the new position of the jumping particle right after the jump.
- The state of inhibition of any neuron  $j \neq i$  is increased by a positive value  $W_{i \rightarrow j}$  at time  $t$ .

In between successive jumps of the system, each neuron  $i$  follows the deterministic dynamic

$$\dot{x}_t^i(x^i) = -\alpha_i(x_t^i(x^i)), \quad x_0^i = x^i, \quad (2.1)$$

where  $\alpha_i$  is positive on  $(0, \infty)$ , locally Lipschitz on  $[0, \infty)$  and  $\alpha_i(0) = 0$  such that the process can not enter the negative values. Let  $\beta_i$  be a continuous positive and decreasing rate function on  $[0, \infty)$ . We have taken  $\beta_i$  to be decreasing so that the larger  $x_t^i$  is, the lower its probability

of spiking and the smaller  $x_t^i$  is, the higher its probability of spiking.  $x_t^i(x^i)$  designs the solution of the equation (2.1) at time  $t$  starting from  $x^i$  at time 0.

We are thus led to consider the piecewise deterministic Markov process (PDMP)  $X_t^N = (X_t^{1,N}, \dots, X_t^{N,N}) \in \mathbb{R}_+^N$ . For  $i \in \{1, \dots, N\}$ , the state of inhibition of neuron  $i$  at time  $t$ ,  $X_t^{i,N}$  is given by:

$$X_t^{i,N} = X_0^{i,N} - \int_0^t \alpha_i(X_{s-}^{i,N}) ds + \int_0^t \int_0^\infty \int_0^\infty (y^i - X_{s-}^{i,N}) \mathbb{1}_{\{r \leq \beta_i(X_{s-}^{i,N})\}} M^i(ds, dr, dy^i) + \sum_{j \neq i} W_{j \rightarrow i} \int_0^t \int_0^\infty \int_0^\infty \mathbb{1}_{\{r \leq \beta_j(X_{s-}^{j,N})\}} M^j(ds, dr, dy^j), \quad (2.2)$$

where  $M^i$  is a random Poisson measure with intensity  $dt dr F^i(dy)$  and for all  $i$ , the  $M^i$  are all independent. This model extends that of Goncalves et al. [12] in the multidimensional case.

*Remark 2.1.* For all  $i \in \{1, \dots, N\}$ ,  $X_t^{i,N}$  can be interpreted as the inhibition state of the neuron  $i$  at time  $t$  and  $W_{j \rightarrow i}$  as the inhibition weight of the neuron  $j$  on the neuron  $i$ . When  $W_{i \rightarrow j} \leq 0$ , we say that the neuron  $i$  is excitatory for the neuron  $j$  and when  $W_{i \rightarrow j} \geq 0$ , we say that the neuron  $i$  is inhibitory for the neuron  $j$ . In our paper we are interested in the case where neuron  $i$  is inhibitory for neuron  $j$  i.e.,  $W_{i \rightarrow j} \geq 0$ .

*Remark 2.2.* The formula (2.2) is well-posed in the sense that there is non explosion of the process. Since  $\beta_i(X_s^{i,N}) \leq \beta_i(0)$  for all  $i$  we deduce that  $\int_0^t \beta_i(X_s^{i,N}) ds < \infty$  whence the non explosion, that is, almost surely, the process has only a finite number of jumps within each finite time interval.

The infinitesimal generator associated with this model is given by:

$$G^N V(x) = - \sum_{i=1}^N \alpha_i(x^i) \frac{\partial}{\partial x^i} V(x) + \sum_{i=1}^N \beta_i(x^i) \int_0^\infty F^i(dy^i) \left[ V(x + e^i y^i - e^i x^i + \sum_{j \neq i} e^j W_{i \rightarrow j}) - V(x) \right] \quad (2.3)$$

where  $V$  is a smooth function and  $e^i$  is the  $i$ -th unit vector.

In other words, at each jump of the process, a single neuron spikes. If it is neuron  $i$  then its state is replaced by  $Y^i$  and all other neurons receive the inhibition weight  $W_{i \rightarrow j} \geq 0$  for any  $j \neq i$ .

## 2.2. First jump time

Let  $N_t^i$  be the counting process of successive jumps of neuron  $i$ , that is,

$$N_t^i = \int_0^t \int_{\mathbb{R}_+} \int_{\mathbb{R}_+} \mathbb{1}_{\{r \leq \beta_i(X_{s-}^{i,N}(x^i))\}} M^i(ds, dr, dy^i)$$

and  $S_1^i$  the first spike time of neuron  $i$ , so we have

$$S_1^i = \inf\{t > 0, N_t^i = 1\}.$$

Let  $S_1$  be the first jump time of the state process  $(X_t)$ , that is,  $S_1 = \min_i S_1^i$ . Let for all  $i$ ,

$$t_0(x^i) := \int_0^{x^i} \frac{dy}{\alpha_i(y)}$$

the time for the neuron  $i$  to hit 0 starting from  $x^i$ .

**Proposition 2.3.** For  $t < \min_i t_0(x^i)$ ,

$$\mathbb{P}(S_1 > t) = \prod_{i=1}^N e^{-[\Gamma_i(x^i) - \Gamma_i(x_t^i(x^i))]}, \quad (2.4)$$

with  $\Gamma_i(x^i) := \int^{x^i} \gamma_i(y) dy$  and  $\gamma_i(x^i) = \beta_i(x^i)/\alpha_i(x^i)$ .

*Proof.* For all  $t > 0$ ,

$$\mathbb{P}(S_1 > t) = \mathbb{P}\left(\min_i S_1^i > t\right) = \prod_{i=1}^N e^{-\int_0^t \beta_i(x_s^i(x^i)) ds}. \quad (2.5)$$

Moreover, if  $t < \min_i t_0(x^i)$  by making a change of variables  $z = x_s^i(x^i)$  and using the fact that  $dz = \dot{x}_s^i(x^i) ds = -\alpha_i(z) ds$  we have

$$\begin{aligned} \mathbb{P}(S_1 > t) &= \prod_{i=1}^N e^{-\int_0^t \beta_i(x_s^i(x^i)) ds} = \prod_{i=1}^N e^{-\int_{x_t^i(x^i)}^{x^i} \beta_i(z) (-dz/\alpha_i(z))} \\ &= \prod_{i=1}^N e^{-[\Gamma_i(x^i) - \Gamma_i(x_t^i(x^i))]} \quad \square \end{aligned}$$

**Assumption 2.4.**  $\Gamma_i(0) = -\infty$  for all  $1 \leq i \leq N$ .

**Proposition 2.5.**

- (1) Suppose Assumption 2.4 holds. Then  $S_1 < \infty$  almost surely.
- (2) Suppose Assumption 2.4 does not hold.
  - If there exists  $i \in \llbracket 1, N \rrbracket$ , such that  $t_0(x^i) < \infty$  then  $S_1 < \infty$  almost surely if and only if  $\beta_i(0) > 0$ .
  - If for all  $i \in \llbracket 1, N \rrbracket$ ,  $t_0(x^i) = \infty$  then  $\mathbb{P}(S_1 = \infty) > 0$  i.e. with a positive probability the first jump time is infinite.

*Proof.*

(1). Let  $N$  be fixed and suppose Assumption 2.4 holds.

- If for all  $i \in \llbracket 1, N \rrbracket$ ,  $t_0(x^i) = \infty$  and letting  $t$  tend to  $\infty$  in (2.4) we have

$$\mathbb{P}(S_1 = \infty) = \prod_{i=1}^N e^{-[\Gamma_i(x^i) - \Gamma_i(x_\infty^i(x^i))]} = \prod_{i=1}^N e^{-[\Gamma_i(x^i) - \Gamma_i(0)]},$$

since  $x_\infty^i(x^i) = 0$  for all  $i \in \llbracket 1, N \rrbracket$ . Then by Assumption 2.4,  $\mathbb{P}(S_1 = \infty) = 0$  that is  $S_1 < \infty$  almost surely.

- If for some  $i \in \llbracket 1, N \rrbracket$ ,  $t_0(x^i) < \infty$  and letting  $t \uparrow \min_i t_0(x^i)$  in (2.4), we obtain

$$\mathbb{P}(S_1 \geq \min_i t_0(x^i)) = \lim_{t \uparrow \min_i t_0(x^i)} \mathbb{P}(S_1 > t) = \prod_{i=1}^N e^{-[\Gamma_i(x^i) - \Gamma_i(0)]} = 0$$

by Assumption 2.4, implying that  $S_1 < \infty$  almost surely.

(2). Suppose Assumption 2.4 does not hold.

- If there exists  $i \in \llbracket 1, N \rrbracket$ , such that  $t_0(x^i) < \infty$  i.e. the time for neuron  $i$  to hit 0 starting from  $x^i$  is finite then it is obvious (by definition of  $t_0(x^i)$ ) to see that it is sufficient for  $\beta_i(0) > 0$  to have  $S_1 < \infty$  almost surely.

## NEURONAL NETWORK

- If Assumption 2.4 does not hold and for all  $i \in \llbracket 1, N \rrbracket$ ,  $t_0(x^i) = \infty$  then by making  $t \rightarrow \infty$  in (2.4) we have  $\mathbb{P}(S_1 = \infty) > 0$  that is  $S_1 = \infty$  with a positive probability.  $\square$

We finish this section with a simulation of the process starting from some fixed initial configuration  $(x_0^1, \dots, x_0^N)$ . For this, we assume that for all  $i$  the jump rate  $\beta_i(x^i)$  is bounded and lower bounded, that is,  $\beta_i(x^i) \in [\beta_*, \beta^*]$  for all  $x^i > 0$ , where  $0 < \beta_* < \beta^* < \infty$ .

The following variables will be used to write our simulation algorithm.

- $T = (T_1, T_2, \dots)$  where  $T_1 < T_2 < \dots$  are the times of successive proposals of jumps for the total system, to be accepted or rejected
- $L$  is the label associated with  $T$ . It will be  $\{sure\}$  or  $\{possible\}$
- $P = (P^1, \dots, P^N)$  is the vector of states of the  $N$  neurons at a fixed instant
- $I$  is the vector which represents the index of the neuron which spikes.

### Algorithm 2.6.

(1) We set  $T_1 \sim \exp(\beta^* N)$

- $L_1 = \{possible\}$  with probability  $\frac{\beta^* - \beta_*}{\beta^*}$
- $L_1 = \{sure\}$  with probability  $\frac{\beta_*}{\beta^*}$

(2) We initialize the vector  $P$  with the values  $(x_0^1, \dots, x_0^N)$

(3) We choose  $I_1 = k$  with probability  $\frac{1}{N}$

- If  $L_1 = \{sure\}$ ,

$$(a) \begin{cases} P^k \sim F^k, \\ P^i \leftarrow x_{T_1}^i(P^i) + W_{k \rightarrow i} \end{cases}$$

- If  $L_1 = \{possible\}$  we accept the jump with probability

$$p = \frac{\beta_k(x_{T_1}^k(P^k)) - \beta_*}{\beta^* - \beta_*},$$

and we apply (a).

- Else  $P^j \leftarrow x_{T_1}^j(P^j)$ ,  $\forall j \in \{1, \dots, N\}$ .

(4) We update the vector  $P$  and start the procedure again from (1).

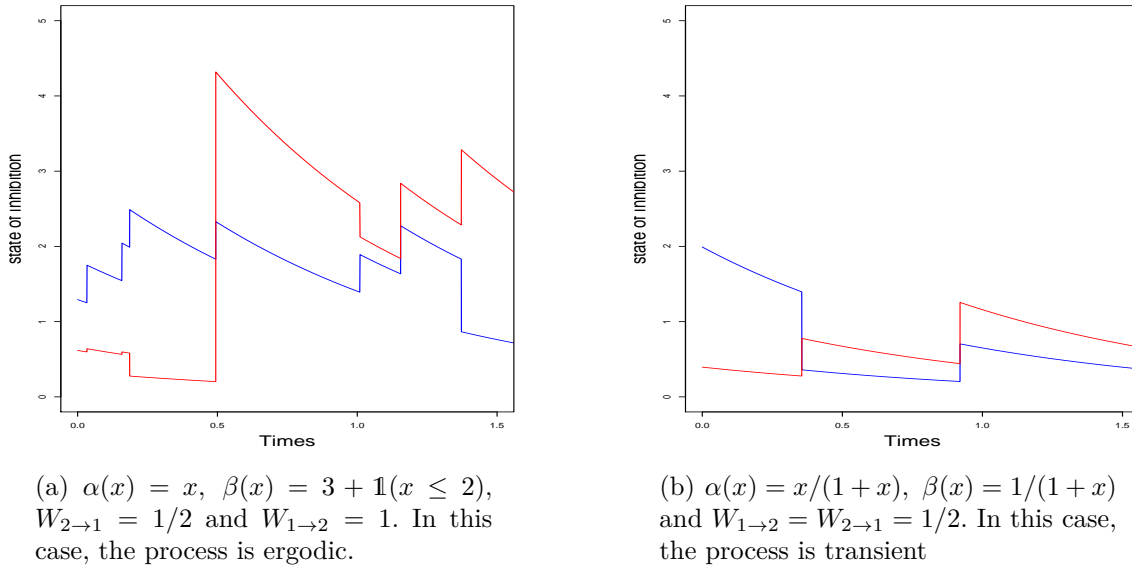
We stop the procedure after a fixed finite number  $n$  of iterations.

We plot in the following figure a typical trajectory of  $X_t^{i,N}$  with  $N = 2$  neurons.

In both figures  $N = 2$  neurons,  $n = 50$  iterations and  $F^i \sim \exp(1)$  for all  $1 \leq i \leq 2$ . It can be seen that in Figure 2.1(a) there are more occurrences of jumps than in Figure 2.1(b). In both, Neuron  $i = 1$  is plotted in blue and neuron  $i = 2$  in red.

### 3. Foster–Lyapunov and Doeblin conditions

In this section, we want to find conditions of non-evanescence of the process and show the existence of an invariant probability measure of the process.


 FIGURE 2.1. Simulations of the neuron system with  $N = 2$  neurons

### 3.1. Foster–Lyapunov condition

In what follows,  $K$  is a fixed compact set, and we suppose that  $\gamma_i(x) := \beta_i(x)/\alpha_i(x)$  is such that

$$\|\gamma_i\|_{\infty, K^c} := \sup_{x \in K^c} |\gamma_i(x)| < \infty.$$

We define  $W$  the matrix of inhibition weight by  $W_{ij} := W_{j \rightarrow i}$ ,  $i \neq j$  and  $W_{ii} = 0$ .

It is further assumed that the matrix  $W$  is irreducible in the sense that there exists an integer  $p > 0$  such that  $W^p$  has only positive coefficients. We introduce the reproduction matrix

$$H_{ij} = W_{j \rightarrow i} \|\gamma_i\|_{\infty, K^c}, \quad i \neq j, \quad H_{ii} = \|\gamma_i\|_{\infty, K^c} \int_0^\infty y^i F^i(dy^i)$$

which is also irreducible.

Suppose that

$$\rho(H) < 1$$

where  $\rho(H)$  is the largest eigenvalue of  $H$ , that is, the spectral radius of  $H$ . By the Perron Frobenius theorem, there exists a left eigenvector  $\kappa$  associated to this eigenvalue  $\rho$ , that is, for all  $i$ ,

$$\sum_j \kappa_j H_{ji} = \rho \kappa_i.$$

On the other hand, put  $m_i = \kappa_i \|\gamma_i\|_{\infty, K^c}$ .

Finally, let  $V : \mathbb{R}_+^N \rightarrow \mathbb{R}_+$  such that

$$V(x) = \sum_{i=1}^N m_i x^i \tag{3.1}$$

and recall that the infinitesimal generator is given by:

$$\begin{aligned} G^N V(x) = & - \sum_{i=1}^N \alpha_i(x^i) \frac{\partial}{\partial x^i} V(x) \\ & + \sum_{i=1}^N \beta_i(x^i) \int_0^\infty F^i(dy^i) \left[ V(x + e^i y^i - e^i x^i + \sum_{j \neq i} e^j W_{i \rightarrow j}) - V(x) \right] \end{aligned}$$

So by replacing  $V$  by its expression in the infinitesimal generator  $G^N V(x)$  we have for all  $x \in K^c$ :

$$\begin{aligned} G^N V(x) &= -\sum_{i=1}^N \alpha_i(x^i) m_i + \sum_{i=1}^N \beta_i(x^i) \int_0^\infty dF^i(y^i) \left[ \sum_{j=1, j \neq i}^N (W_{i \rightarrow j} + x^j) m_j + y^i m_i - \sum_{j=1}^N x^j m_j \right] \\ &= -\sum_{i=1}^N \alpha_i(x^i) m_i + \sum_{i=1}^N \beta_i(x^i) \left( m_i \int_0^\infty y^i F^i(dy^i) + \sum_{j \neq i} W_{i \rightarrow j} m_j \right) - \sum_{i=1}^N \beta_i(x^i) x^i m_i. \end{aligned}$$

Then, since  $-\beta_i(x^i) x^i \leq 0$ ,

$$\begin{aligned} G^N V(x) &\leq -\sum_i \alpha_i(x^i) m_i + \sum_i \beta_i(x^i) \left( m_i \int_0^\infty y^i F^i(dy^i) + \sum_{j \neq i} W_{i \rightarrow j} m_j \right) \\ &= -\sum_i \alpha_i(x^i) \left( m_i - \gamma_i(x^i) \left[ \frac{m_i}{\|\gamma_i\|_{\infty, K^c}} H_{ii} + \sum_{j \neq i} \frac{1}{\|\gamma_j\|_{\infty, K^c}} H_{ji} m_j \right] \right) \\ &= -\sum_i \alpha_i(x^i) \left( m_i - \gamma_i(x^i) \left[ \kappa_i H_{ii} + \sum_{j \neq i} \kappa_j H_{ji} \right] \right) \\ &= -\sum_i \alpha_i(x^i) \left( \|\gamma_i\|_{\infty, K^c} \kappa_i - \gamma_i(x^i) \rho \kappa_i \right) \\ &= -\sum_i \alpha_i(x^i) \|\gamma_i\|_{\infty, K^c} \kappa_i \left( 1 - \frac{\gamma_i(x^i)}{\|\gamma_i\|_{\infty, K^c}} \rho \right). \end{aligned}$$

This calculus leads us to introduce the following

**Assumption 3.1.** Let  $\alpha > 0$ . For all  $i$ , there exists  $r_i$ , such that  $\forall x^i \geq r_i$ ,  $\alpha_i(x^i) \geq \alpha x^i$ .

**Corollary 3.2.** Under Assumption 3.1 we have for all  $x \in K^c$  such that  $x^i \geq r_i$  for all  $i$  and such  $\alpha_i(x^i) \geq \alpha x^i$  :

$$G^N V(x) \leq -\alpha \sum_i x^i m_i \left( 1 - \frac{\gamma_i(x^i)}{\|\gamma_i\|_{\infty, K^c}} \rho \right) \leq -cV(x)$$

where  $c$  is a positive constant.

**Definition 3.3.** We call the process non evanescent if there exists a compact  $K$  such that for all  $x$ ,  $\mathbb{P}_x$ - almost surely,  $\limsup_t 1_K(X_t) = 1$ .

**Proposition 3.4.** If  $\rho < 1$ , then the process is non-evanescent.

*Proof.*  $V(x)$  defined in (3.1) above is a norm-like function because the eigenvector  $\kappa$  is positive. Indeed, we call  $V : \mathbb{R}_+^N \rightarrow \mathbb{R}$  a norm-like function if  $V$  is a positive, measurable function and  $V(x) \rightarrow \infty$  when  $x \rightarrow \infty$ . The condition (CD1) of Meyn and Tweedie [14] implies the result.  $\square$

*Example 3.5* (Mean-field interaction). Suppose we have  $N$  neurons. We suppose also the function  $\gamma_i$  such that  $\|\gamma_i\|_{\infty, K^c} < \infty$  and  $F^i = F$ ,  $W_{j \rightarrow i} = \theta$  for all  $i$ . In this case the reproduction matrix is

$$H_{ij} = \theta \|\gamma_i\|_{\infty, K^c}, \quad i \neq j, \quad H_{ii} = \mathbb{E}(Y) \|\gamma_i\|_{\infty, K^c}$$

for some fixed compact set  $K$ . Suppose  $\rho(H)$  is the spectral radius of  $H$ . Then,  $\rho(H) = \sup_{x \in K^c} \gamma_i(x) (\mathbb{E}(Y) + (N-1)\theta)$  and its associated eigenvector is  $\kappa = (1, \dots, 1)$ . The condition  $\rho(H) < 1$  is therefore equivalent to  $\sup_{x \in K^c} \gamma_i(x) < 1/(\mathbb{E}(Y) + (N-1)\theta)$ .

*Example 3.6* (Torus). Suppose we have  $N \geq 3$  neurons such that each neuron interacts with its two nearest neighbors (its left and right neighbors). Neuron 1 interacts with neuron 2 and neuron  $N$ . Neuron  $N$  interacts with neuron  $N-1$  and neuron 1, so we have a torus.



We suppose also  $\gamma_i$  such that  $\|\gamma_i\|_{\infty, K^c} < \infty$  and  $F^i = F$ ,  $W_{j \rightarrow i} = \theta$  for all  $j \in \{i+1, i-1\}$  and  $W_{j \rightarrow i} = 0$  if  $j \notin \{i+1, i-1\}$ . In this case the reproduction matrix is

$$H_{ij} = \begin{cases} \theta \|\gamma_i\|_{\infty, K^c} & \text{if } i \neq j, j \in \{i+1, i-1\} \\ 0 & \text{if } i \neq j, j \notin \{i+1, i-1\} \\ \mathbb{E}(Y) \|\gamma_i\|_{\infty, K^c} & \text{if } i = j. \end{cases}$$

If  $\rho(H)$  is the spectral radius of  $H$  then  $\rho(H) = \|\gamma_i\|_{\infty, K^c} (\mathbb{E}(Y) + 2\theta)$  and its associated eigenvector is  $\kappa = (1, \dots, 1)$ . The condition  $\rho(H) < 1$  is equivalent to  $\|\gamma_i\|_{\infty, K^c} (\mathbb{E}(Y) + 2\theta) < 1$ .

### 3.2. Doeblin condition

Let  $S_0 < S_1 < \dots < S_n < \dots$  be the instants of successive jumps of the  $N$  neurons. It is obvious that the embedded chain  $Z_n := X_{S_n}$  is a Markov chain. Let  $I_n$  be the index of the neuron which jumps at time  $S_n$ .

**Proposition 3.7.** *Suppose that the assumptions of Proposition 2.5 hold. Then,  $(Z_n, I_n)$  is a Markov chain and its transition  $Q(x, dy)$  is given by:*

$$\begin{aligned} \mathbb{P}(Z_n \in dy, I_n = j | Z_{n-1} = x, I_{n-1} = i) &= \int_0^\infty ds e^{-\int_0^s dt \sum_{i=1}^N \beta_i(x_i^i(x^i))} \beta_j(x_s^j(x^j)) \\ &\times \int F^j(du) \delta_{(x_s^1(x^1) + W_{j \rightarrow 1}, \dots, x_s^{j-1}(x^{j-1}) + W_{j \rightarrow j-1}, u, x_s^{j+1}(x^{j+1}) + W_{j \rightarrow j+1}, \dots, x_s^N(x^N) + W_{j \rightarrow N})}(dy). \end{aligned} \quad (3.2)$$

**Theorem 3.8.** *Suppose for all  $1 \leq i \leq N$ ,  $\alpha_i \in \mathcal{C}^1$  and there exists a compact set  $K \subset (0, \infty)^N$  such that for all  $x \in K$ , for all  $1 \leq i \leq N$ ,  $\beta_i(x^i + \sum_{j=1}^{i-1} W_{j \rightarrow i}) > 0$ . Moreover we suppose that  $F^i(dy)$  is absolutely continuous and  $\|\beta_i\|_\infty < \infty$  for all  $i$ . Then there exist  $d \in (0, 1)$  and a probability measure  $\nu$  on  $(\mathbb{R}_+^N, \mathcal{B}(\mathbb{R}_+^N))$ , such that*

$$Q^N(x, dy) \geq d 1_K(x) \nu(dy) \quad (3.3)$$

where  $Q$  is the transition operator of embedded chain  $Z_n = X_{S_n}$  and  $Q^N$  is its  $N$ -th iterate.

To prove the above result we fix any deterministic sequence  $s_1 < \dots < s_N$ . In the sequel we shall work on the event  $S_1 = s_1, \dots, S_N = s_N, I_1 = 1, \dots, I_N = N$  and  $Y_1 = y_1, \dots, Y_N = y_N$ . This means that the jumps are ordered such that neuron 1 jumps before neuron 2 and etc. Let  $y = (y^1, \dots, y^N)$  where  $y^i$  is the new state of inhibition of neuron  $i$  after the spike.

Let  $t_k = s_k - s_{k-1}$  for all  $1 \leq k \leq N$  the inter jump times of the  $N$  neurons which implies that  $s_k = t_1 + \dots + t_k$ .

Conditionally on this event, let  $\Psi_{s_N}$  be the vector of states of the process at time  $s_N$ . We can define  $\Psi_{s_N}$  as a function of the states  $y_1, \dots, y_N$  such that  $\Psi_{s_N} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  is given by:

$$\Psi_{s_N}^k(y) = \begin{cases} \psi_{t_N}^{k,N} \circ \dots \circ \psi_{t_{k+1}}^{k,k+1}(y^k) & \text{if } 1 \leq k < N \\ y^N & \text{if } k = N \end{cases}$$

where for all  $l \neq k$ ,

$$\psi_s^{k,l}(u) = x_s^k(u) + W_{l \rightarrow k} \quad (3.4)$$

and  $x_s^k(u)$  means the solution of the deterministic dynamic  $\dot{x}_s^k = -\alpha_k(x_s^k)$ ,  $x_0^k = u$ .

*Remark 3.9.* In the definition of  $\Psi_{s_N}^k(y)$ , we note that it depends only on  $y^k$ . Therefore we have for all  $i \neq j$ ,

$$\frac{\partial \Psi_{s_N}^i}{\partial y^j} = 0.$$

**Proposition 3.10.** *For all  $1 \leq k \leq N$  let  $\alpha_k$  be a globally Lipschitz function. For all  $y \in \mathbb{R}_+^N$ , there exists an open neighborhood  $\mathcal{B}$  of  $y$  such that  $\Psi_{s_N} : \mathcal{B} \rightarrow \Psi_{s_N}(\mathcal{B})$  is a local diffeomorphism.*

## NEURONAL NETWORK

*Proof.* Let  $J_{\Psi_{s_N}(y)}$  be the Jacobian matrix of  $\Psi_{s_N}(y)$ . Using Remark 3.9 we have :

$$\det(J_{\Psi_{s_N}(y)}) = \det \begin{pmatrix} \frac{\partial \Psi_{s_N}^1(y)}{\partial y^1} & \cdots & \frac{\partial \Psi_{s_N}^1(y)}{\partial y^N} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Psi_{s_N}^N(y)}{\partial y^1} & \cdots & \frac{\partial \Psi_{s_N}^N(y)}{\partial y^N} \end{pmatrix} = \det \begin{pmatrix} \frac{\partial \Psi_{s_N}^1(y)}{\partial y^1} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\partial \Psi_{s_N}^N(y)}{\partial y^N} \end{pmatrix}.$$

We obtain  $\det(J_{\Psi_{s_N}(y)}) \neq 0$  if and only if  $\prod_{j=1}^N \frac{\partial \Psi_{s_N}^j(y)}{\partial y^j} \neq 0$  that is  $\frac{\partial \Psi_{s_N}^j(y)}{\partial y^j} \neq 0, \forall 1 \leq j \leq N$ . It is obvious to see that  $\frac{\partial \Psi_{s_N}^N(y)}{\partial y^N} = 1$ .

For all  $1 \leq j \leq N-1$ , we have:

$$\frac{\partial \Psi_{s_N}^j(y)}{\partial y^j} = \prod_{i=1}^{N-j} \exp \left( - \int_{s_{N-i}}^{s_{N-(i-1)}} \alpha'_j \left( x_s^j \left( \Psi_{s_{N-i}}^j(y) \right) \right) ds \right) \neq 0. \quad (3.5)$$

It means that  $|\det(J_{\Psi_{s_N}(y)})| \neq 0$  then  $\Psi_{s_N}(y)$  is a local diffeomorphism. Localizing, we may therefore conclude that for each  $y$  there exists  $\mathcal{B}$  such that  $\Psi_{s_N} : \mathcal{B} \rightarrow \Psi_{s_N}(\mathcal{B})$  is a diffeomorphism.  $\square$

*Proof of Theorem 3.8.* Let  $\varepsilon > 0$  fixed. We will work on the event

$$E = \{S_1 \leq \varepsilon, \dots, S_{n+1} - S_n \leq \varepsilon, \forall n < N : (I_1, \dots, I_N) = (1, \dots, N)\}.$$

In particular, on  $E$ , the index  $I_n$  of the  $n$ -th neuron is equal to  $n$  for all  $n \in \{1, \dots, N\}$ .

Knowing that the first jump takes place at time  $S_1 = s_1$ , the probability that the index  $I_1$  of the first jump is equal to 1 is given by:

$$\mathbb{P}(I_1 = 1 | S_1 = s_1) = \mathbb{P}(S_1^1 < S_1^j, \forall j \neq 1) = \frac{\beta_1(x_{s_1}^1(x^1))}{\sum_{j=1}^N \beta_j(x_{s_1}^j(x^j))}.$$

We want to compute,  $\mathbb{P}(I_1 = 1, \dots, I_N = N | S_1 = s_1, S_2 = s_2, \dots, S_N = s_N)$ . To obtain a compact formula, using formula (3.4) we define

$$\phi_j^k(x^k, y^k, s_1, \dots, s_N) = \begin{cases} \psi_{t_j}^{k,j} \circ \dots \circ \psi_{t_{k+1}}^{k,k+1}(y^k) & \text{if } 1 \leq k \leq j-1 \\ \psi_{t_j}^{k,j} \circ \dots \circ \psi_{t_1}^{k,1}(x^k) & \text{if } j \leq k \leq N \end{cases}$$

giving the states of neuron  $k$  at time  $S_j$  depending on whether neuron  $k$  jumped before or after time  $S_j$ . Let

$$x_j^k = x_{t_j}^k(\phi_{j-1}^k(x^k, y^k, s_1, \dots, s_N))$$

be the state of neuron  $k$  before the  $j$ -th jump. We know that as long as neuron  $k$  has not yet jumped, it receives each time a quantity  $W_{j \rightarrow k}, \forall j \neq k$  from the other neurons that jumped before it. So knowing all the jump times where other neurons jumped, we have:

$$\begin{aligned} \mathbb{P}(I_1 = 1, \dots, I_N = N | S_1 = s_1, S_2 = s_2, \dots, S_N = s_N) \\ = \frac{\beta_1(x_{s_1}^1(x^1))}{\sum_{i=1}^N \beta_i(x_{s_1}^i(x^i))} \int_{\mathbb{R}_+^{N-1}} \frac{\prod_{i=2}^N \beta_i(x_i^i)}{\prod_{i=2}^N (\sum_{k=1}^N \beta_k(x_i^k))} \prod_{k=1}^{N-1} \mathbb{P}(Y^k \in dy^k). \end{aligned}$$

For any Borel subset  $B$  of  $\mathbb{R}^N$  we have

$$\begin{aligned} Q^N(x, B) \geq \mathbb{P}_x(Z_N \in B, E) &= \int_{[0, \varepsilon]^N} dt_1 \dots dt_N \int_{\mathbb{R}^N} F^1(dy^1) \dots F^N(dy^N) \\ &\times \left( \prod_{k=1}^N \beta_k(x_k^k) \right) e^{-\int_0^{s_N} \sum \beta_k(\Psi_t^k(y)) dt} \mathbf{1}_B(\Psi_{s_N}(y)). \end{aligned}$$

Remark that on the event  $E$ ,  $x_k^k \leq x^k + \sum_{j=1}^{k-1} W_{j \rightarrow k}$ . Recall  $\beta_k$  is decreasing function and let  $\mu_k = \inf_k \{\beta_k(x^k + \sum_{j=1}^{k-1} W_{j \rightarrow k}) : x \in K\}$  the lower-bound on  $K$  of  $\beta_k(x^k + \sum_{j=1}^{k-1} W_{j \rightarrow k})$ .

Using the fact that  $\|\beta_i\|_\infty < \infty$  for all  $i$ , let  $c = (\prod_{k=1}^N \mu_k) e^{-N\|\beta_i\|_\infty N\varepsilon}$ . Then we have

$$Q^N(x, B) \geq c \int_{[0, \varepsilon]^N} dt_1 \dots dt_N \int_{\mathbb{R}^N} F^1(dy^1) \dots F^N(dy^N) \mathbf{1}_B(\Psi_{s_N}(y)). \quad (3.6)$$

Following the arguments of Benaïm et al. [3], for any  $t^* \leq N\varepsilon$ , there exists a ball  $B_r(t^*)$  of radius  $r$ , of center  $t^*$  and an open set  $I \subset \mathbb{R}^N$  such that we can find for all  $s_N \in B_r(t^*)$ , an open set  $W_{s_N} \subset \mathbb{R}^N$  :

$$\tilde{\Psi}_{s_N} : \begin{cases} W_{s_N} \rightarrow I \\ y \mapsto \Psi_{s_N}(y) \end{cases}$$

is a diffeomorphism (see Benaïm et al. [3, Lemma 6.2]). In the above formula,  $\tilde{\Psi}_{s_N}$  denotes the restriction of  $\Psi_{s_N}$  to  $W_{s_N}$ . This allows us to apply the theorem of a change of variables in the inequality (3.6).

$\alpha'_j(x_s^j(\Psi_{s_N-i}^j(y)))$  is upper bounded since  $\alpha_j$  is a global Lipschitz function. Then, for all  $1 \leq j \leq N-1$  we obtain:

$$\frac{\partial \Psi_{s_N}^j}{\partial y^j}(y) \leq \exp(N-j)\varepsilon \|\alpha'_j\|_\infty.$$

Then,  $\forall y \in W_{s_N}$ ,  $c |\det(J_{\Psi_{s_N}(y)})|^{-1} \geq c' > 0$  and the inequality (3.6) becomes :

$$\begin{aligned} Q^N(x, B) &\geq c \int_{[0, \varepsilon]^N} dt_1 \dots dt_N \int_B F^1(dy^1) \dots F^N(dy^N) \mathbf{1}_B(\Psi_{s_N}(y)) \\ &\geq c' \int_{B_r(t^*)} dt_1 \dots dt_N \int_{W_{s_N} \cap B} dy \mathbf{1}_B(\tilde{\Psi}_{s_N}(y)) |\det(J_{\tilde{\Psi}_{s_N}(y)})| \\ &\geq c' \lambda(B_r(t^*)) \int_I \mathbf{1}_B(x) dx = d \mathbf{1}_B(x) \nu(I) \end{aligned}$$

where  $d = c' \lambda(B_r(t^*))$  with  $\lambda(B_r(t^*))$  the Lebesgue measure of the ball  $B_r(t^*)$  and  $\nu(I)$  the uniform measure of  $I$ .  $\square$

**Corollary 3.11.** *If for all  $k \leq N$ ,  $\beta_k$  is strictly lower-bounded and bounded, then the process is recurrent.*

*Proof.* When  $\beta_k$  is strictly lower-bounded and bounded, we can notice that the lower bound obtained in Theorem 3.8 holds on the whole state space  $\mathbb{R}_+$ , that is, without  $1_K$ . This allows us to have the global lower bound  $Q^N(x, dy) \geq d\nu(dy)$  and thus the uniform ergodicity of the process.  $\square$

#### 4. Perfect simulation

In this section, we consider a framework with an infinity of neurons indexed by  $\mathbb{Z}$ . We want to build a perfect simulation algorithm to show in another way the recurrence of our process under certain conditions. Let  $\mathcal{V}_{\rightarrow i} = \{j : W_{j \rightarrow i} \neq 0\}$  and  $\mathcal{V}_{i \rightarrow} = \{j : W_{i \rightarrow j} \neq 0\}$  be the incoming and out-coming neighborhoods of the neuron  $i$  (see Comets et al. [4] and Galves and Löcherbach [10]).

We consider the case where each neuron has a finite number of neighbors.

We assume throughout this section that for all  $i$  the jump rate  $\beta_i(x^i)$  is bounded, that is,  $\beta_i(x^i) \in [\beta_*, \beta^*]$  for all  $x^i > 0$ , where  $0 < \beta_* < \beta^* < \infty$ .

The following variables will be used to write the perfect simulation algorithm:

- $T$  is the time vector

## NEURONAL NETWORK

- $P$  is the matrix of states where each row of this matrix represents the different states of the  $N$  neurons at a fixed instant
- $I$  is the vector which represents the index of the neuron which spikes.

We fix a neuron  $i \in \mathbb{Z}$  and in what follows we are interested in finding the state of  $i$  at time 0 in the stationary regime, that is, assuming that the process starts from  $-\infty$ . To do so we explore the past in order to determine all sets of indices and times which affect the value of neuron  $i$  at time 0.

To explain what we mean by this, let us consider the following example where the interactions are given in the case of nearest neighbors. In the following example, the red dots represent possible jumps and the blue stars represent sure jumps. The sure and possible jumps are the same as in Algorithm 2.6.

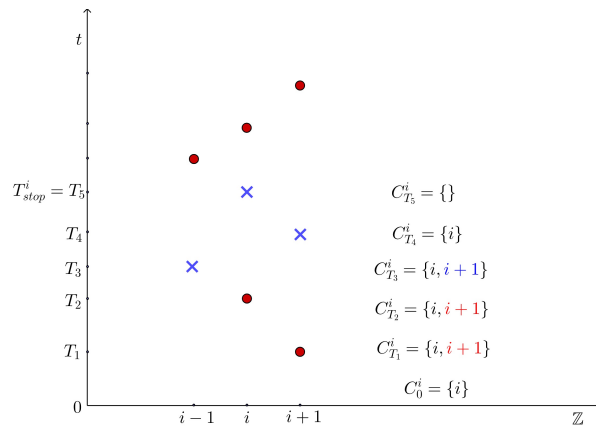


FIGURE 4.1. Clan of ancestors if neuron  $i$

In this example, we have fixed a neuron  $i$  in  $\mathbb{Z}$  at time 0 and we say that the clan of ancestors of neuron  $i$  is reduced to neuron  $i$  itself. It is assumed that the space of neurons is reduced to  $i - 1, i, i + 1$ . Then, at time  $T_1$ , neuron  $i + 1$  makes a possible jump. We record the time  $T_1$  and we add the neuron  $i + 1$  to the clan of the ancestors of the neuron  $i$ . At time  $T_2$  neuron  $i$  makes a possible jump. As neuron  $i$  is already in the ancestor clan then the clan remains unchanged and we download the time  $T_2$ . At time  $T_3$  neuron  $i - 1$  makes a sure jump. We register the time  $T_3$  and the neuron  $i - 1$  but the clan remains unchanged. At time  $T_4$  the neuron  $i + 1$  makes a sure jump and as the neuron  $i + 1$  is already in the clan, we remove from the clan and only the neuron  $i$  remains in the clan. At time  $T_5$  neuron  $i$  makes a sure jump and as neuron  $i$  is already in the clan, we remove from the clan and the clan becomes empty. Our algorithm stops the first time the clan becomes empty. In the following algorithm we will work in a general case.

The set of neurons thus constructed will be called the ancestor clan of neuron  $i$ . (see Galves and Löcherbach [10], Galves et al. [11]). The clan of ancestors is a process that evolves in time by successive jumps. We start with  $C_0^i = \{i\}$  and in the following we will define the updates of this process at the times of the jumps. More precisely we do the following:

**Algorithm 4.1** (Backward procedure).

- (1) We simulate,  $\forall l \in \mathbb{Z}$ ,  $N_t^{l,s}$  and  $N_t^{l,p}$  two Poisson processes with respective intensities  $\beta_*$  and  $\beta^* - \beta_*$ . The jump times of  $N_t^{l,s}$  and  $N_t^{l,p}$  are respectively  $T_n^{l,s}$  and  $T_n^{l,p}$  for the neuron  $l$  after  $n$  jumps.
- (2) Let  $i \in \mathbb{Z}$  be fixed and  $T_1 = \inf\{T_1^{j,r} > 0 : j \in \mathcal{V}_{\rightarrow i}, T_1^{i,r} > 0\}$  where  $r \in \{p, s\}$  and  $\mathcal{V}_{\rightarrow i}$  is the incoming neighborhood of  $i$ .

- If  $T_1 = T_1^{j,p}$ , we set  $C_{T_1}^i = \{i, j\}$  and we set  $I_1 = j$ .
- If  $T_1 = T_1^{j,s}$ , we set  $C_{T_1}^i = \{i\}$  and  $I_1 = j$ .
- If  $T_1 = T_1^{i,p}$ , we set  $C_{T_1}^i = \{i\}$  and we set  $I_1 = i$ .
- If  $T_1 = T_1^{i,s}$ , we set  $C_{T_1}^i = \emptyset$  and we stop the algorithm. In this case we set  $I_1 = i$ .

(3) Suppose  $T_n$  is the  $n$ -th jump time of  $C_{T_n}^i$ . Then,

$$T_{n+1} = \inf\{T_m^{j,s}, T_m^{j,p} > T_n : \exists l \in C_{T_n}^i, j \in \mathcal{V}_{\rightarrow l}, T_m^{k,p}, T_m^{k,s} > T_n, k \in C_{T_n}^i\}.$$

- If  $T_{n+1} = T_m^{j,s}$  we set  $I_{n+1} = j$  and then  $C_{T_{n+1}}^i = C_{T_n}^i$ .
- If  $T_{n+1} = T_m^{j,p}$  we set  $I_{n+1} = j$  and then  $C_{T_{n+1}}^i = C_{T_n}^i \cup \{j\}$ .
- If  $T_{n+1} = T_m^{k,p}$  we set  $I_{n+1} = k$  and then  $C_{T_{n+1}}^i = C_{T_n}^i$ .
- If  $T_{n+1} = T_m^{k,s}$  we set  $I_{n+1} = k$  and then  $C_{T_{n+1}}^i = C_{T_n}^i \setminus \{k\}$  where  $k \in C_{T_n}^i$ .

We stop the procedure at time  $T_{stop}^i = \inf\{t : C_t^i = \emptyset\}$ .

To ensure that the algorithm stops it will be necessary to find a criterion so that  $T_{stop}^i < \infty$ . This will be done in Theorem 4.7 below. The above algorithm is called the backward procedure.

In the following we will write a forward procedure of the process in case where each neuron has a finite number of neighbors and in case  $T_{stop}^i < \infty$ . For this we define:

$$N_{stop}^i = \inf\{n > 0 : C_{T_n}^i = \emptyset\}, \quad \bar{C}^i = \bigcup_{n=0}^{N_{stop}^i} C_{T_n}^i \quad \text{and} \quad \partial_{ext}(C_t^i) = \{j \notin C_t^i : \exists k \in C_t^i, W_{j \rightarrow k} > 0\}$$

where  $N_{stop}^i$  is the number of steps of the backward procedure,  $\bar{C}^i$  is the union of all clans of ancestors up to  $N_{stop}^i$  and  $\partial_{ext}(C_t^i)$  is the set of neurons not belonging to the clan of ancestor of neuron  $i$  but having an interaction with at least one neuron in the ancestor clan of neuron  $i$ .

In this algorithm, we will rely on the a priori realizations of the processes  $N_t^{i,s}$  and  $N_t^{i,p}$ .

**Algorithm 4.2** (Forward procedure).

(1) We initialize the set of sites for which the decision to accept can be made by

$$\mathcal{S}^i = \{(I_m, T_m) \in \bar{C}^i \times \mathbb{R}_+, C_{T_m}^i = \emptyset\}$$

For  $n = N_{stop}^i$  we have  $P_n^{I_n} \sim F^{I_n}$ . Starting from  $n \rightarrow n - 1$ :

(2) If  $(I_{n-1}, T_{n-1}) \in \mathcal{S}^i$  then  $P_{n-1}^{I_{n-1}} \sim F^{I_{n-1}}$ .

- If for  $j \in \mathcal{V}_{I_{n-1} \rightarrow \cdot}$ , we have  $j \in C_{T_{n-1}}^i$  then

$$P_{n-1}^j = x_{T_n - T_{n-1}}^j(P_n^j) + W_{I_{n-1} \rightarrow j}$$

- If for  $j \notin \mathcal{V}_{I_{n-1} \rightarrow \cdot}$ , we have  $j \in C_{T_{n-1}}^i$  then

$$P_{n-1}^j = x_{T_n - T_{n-1}}^j(P_n^j)$$

(3) If  $k := I_{n-1} \in \partial_{ext}(C_{T_{n-1}}^i)$ , we have

$$P_{n-1}^l = x_{T_n - T_{n-1}}^l(P_n^l) + W_{k \rightarrow l}$$

where there exists  $l$  such that  $k \rightarrow l \in C_{T_{n-1}}^i$

## NEURONAL NETWORK

(4) If  $(I_{n-1}, T_{n-1}) \in (\bar{C}^i \times \mathbb{R}_+) \setminus \mathcal{S}^i$  then:

- We decide according to the probabilities

$$p = \frac{\beta_{I_{n-1}}(x_{T_n - T_{n-1}}^{I_{n-1}}(P_n^{I_{n-1}})) - \beta_*}{\beta^* - \beta_*}$$

to accept the presence of a spike of neuron  $I_{n-1}$ .

We update

$$\mathcal{S}^i \leftarrow \mathcal{S}^i \cup \{(I_m, T_m) \in \bar{C}^i \times \mathbb{R}_+, C_{T_m}^{I_m} \subset \mathcal{S}^i\}$$

and go back to Step 2.

- Else with the probabilities  $1 - p$  we reject the presence of a spike of neuron  $I_{n-1}$  and  $P_{n-1}^{I_{n-1}} = x_{T_n - T_{n-1}}^{I_{n-1}}(P_n^{I_{n-1}})$ .

We consider all the elements of  $\mathcal{S}^i$  and we always start with the last element to get out of the clan. The update of  $\mathcal{S}^i$  allows us to start the procedure again.

We stop the procedure when all the elements of  $\bar{C}^i$  are filled.

*Remark 4.3.*

- The output of the above algorithm 4.2 is a sample of the process in its stationary state. For more on this, see [9, p. 21].
- For any site  $(i, t) \in \mathbb{Z} \times \mathbb{R}_+$ ,  $C_t^i$  is a Markov jump process taking values in the finite subset of  $\mathbb{Z}$  (see Galves et al. [11]) and its infinitesimal generator is given by

$$A^{clan}g(C) = \sum_{j \in C} \beta_* [g(C \setminus \{j\}) - g(C)] + \sum_{j \in \partial_{ext}(C)} (\beta^* - \beta_*) [g(C \cup \{j\}) - g(C)]$$

where  $g$  is a test function.

**Proposition 4.4.** Let  $d_j = \min_{x^j} \beta_j(x^j)$ ,  $d^j = \max_{x^j} \beta_j(x^j)$  and  $b_j = \sum_{k \rightarrow j} (d^k - d_k)$  where  $\sum_{k \rightarrow j}$  means the sum over all neurons  $k$  such that  $W_{k \rightarrow j} \neq 0$ . If  $\sup_j b_j < \inf_j d_j$  then for all  $j$ ,  $T_{stop}^j$  is finite almost surely.

*Proof.* Let  $i$  be a fixed neuron and  $C_t^i$  the clan of ancestors of neuron  $i$  at time  $t$ . We set  $\bar{b} = \sup_j b_j$  and  $\underline{d} = \inf_j d_j$ . We shall construct a process  $Z = (Z_t)_t$  such that for all  $n$ ,  $|C_{T_n}^i| \leq |\bar{Z}_n|$  where  $\bar{Z}_n = Z_{T_n}$ .

We proceed as follows:

- (1) The neurons of the clan of ancestors  $C_t^i$  jump up with jump rate  $\sum_{j \in C_t^i} b_j \leq \bar{b}|C_t^i|$  and jump down with jump rate  $\sum_{j \in C_t^i} d_j \geq \underline{d}|C_t^i|$ .
- (2) When we add an element of clan  $C_t^i$ , we also add an element of  $Z_t$ . And when we remove an element from  $Z_t$ , we also remove an element of clan  $C_t^i$ . But we can remove an element of clan  $C_t^i$  and not of  $Z_t$ , so the two processes do not always jump together.

Therefore we may couple  $C_t^i$  with a classical birth and death chain having birth rate  $\bar{b}|\bar{Z}_n|$  and death rate  $\underline{d}|\bar{Z}_n|$ .

We notice that  $\mathbb{E}(\bar{Z}_1) = 2\bar{b}/(\bar{b} + \underline{d})$ , then if  $\bar{b} < \underline{d}$  then almost surely  $\lim_{n \rightarrow \infty} \bar{Z}_n = 0$  (see for instance, Theorem 1 of Athreya and Ney [2]). Which implies that  $\lim_{n \rightarrow \infty} |C_{T_n}^i| = 0$  this implies  $T_{stop}^i$  is finite almost surely.  $\square$

In this general case where a neuron has a finite number of neighbors (more than two neighbors) with which it interacts, we can say no more than Proposition 4.4. Thus, in the following, we put ourselves in the case where each neuron  $i$  has exactly two neighbors so that the neuron  $i$  interacts only with the neurons  $i + 1$  and  $i - 1$ . In other words, the incoming neighborhood of  $i$  is  $\mathcal{V}_{\rightarrow i} = \{i + 1, i - 1\}$ .

**Algorithm 4.5** (Backward procedure).

(1) We simulate,  $\forall l \in \mathbb{Z}$ ,  $N_t^{l,s}$  and  $N_t^{l,p}$  two Poisson processes with respective intensities  $\beta_*$  and  $\beta^* - \beta_*$ . The jump times of  $N_t^{l,s}$  and  $N_t^{l,p}$  are respectively  $T_n^{l,s}$  and  $T_n^{l,p}$  for the neuron  $l$  after  $n$  jumps. The jump times  $T_n^{l,s}$  will be considered as times of sure jumps (counted by the process  $N_t^{l,s}$ ) and the jump times  $T_n^{l,p}$  will be considered as times of possible jumps (counted by the process  $N_t^{l,p}$ )

(2) Let  $i \in \mathbb{Z}$  fix and  $T_1 = \inf\{T_1^{i\pm 1,s}, T_1^{i\pm 1,p}, T_1^{i,s}, T_1^{i,p}\}$ .

- If  $T_1 = T_1^{i\pm 1,p}$ , we set  $C_{T_1}^i = \{i, i \pm 1\}$ . We put  $I_1 = i \pm 1$ .
- If  $T_1 = T_1^{i\pm 1,s}$ , we set  $C_{T_1}^i = i$  and we put  $I_1 = i \pm 1$ .
- If  $T_1 = T_1^{i,p}$ , we set  $C_{T_1}^i = i$  and we put  $I_1 = i$ .
- If  $T_1 = T_1^{i,s}$ , we set  $C_{T_1}^i = \emptyset$  and we stop the algorithm. We put  $I_1 = i$ .

(3) Suppose  $T_n$  is the  $n$ -th jump time of  $C_{T_n}^i$ . We have:

$$T_{n+1} = \inf\{T_m^{j,s}, T_m^{j,p} > T_n : |j - C_{T_n}^i| \leq 1, T_m^{k,p}, T_m^{k,s} > T_n, k \in C_{T_n}^i\}.$$

- If  $T_{n+1} = T_m^{j,p}$  we set:

$$\begin{cases} \text{If } j \in C_{T_n}^i, & C_{T_{n+1}}^i = C_{T_n}^i \\ \text{If } j \notin C_{T_n}^i, & C_{T_{n+1}}^i = C_{T_n}^i \cup \{j\} \end{cases}$$

- If  $T_{n+1} = T_m^{k,s}$  we set:

$$\begin{cases} \text{If } k \in C_{T_n}^i, & C_{T_{n+1}}^i = C_{T_n}^i \setminus \{k\} \\ \text{If } k \notin C_{T_n}^i, & C_{T_{n+1}}^i = C_{T_n}^i \end{cases}$$

We update  $C_t^i$  and start the procedure again. We stop the procedure at time  $T_{stop}^i$  where  $T_{stop}^i = \inf\{t : C_t^i = \emptyset\}$ .

Indeed, the whole procedure makes sense only if  $T_{stop}^i < \infty$  almost surely.

*Remark 4.6.* The forward procedure is the same as in the first case where each neuron has a finite number of neighbors.

The following theorem gives conditions on the extinction time of the process.

**Theorem 4.7.** We set  $\delta = \frac{\beta_*}{\beta^* - \beta_*}$ . There exists a critical value  $0 < \delta_c < \infty$  such that:

- if  $\delta > \delta_c$ , then the extinction time is finite almost surely that is,  $\mathbb{P}(\forall i, T_{stop}^i < \infty) = 1$
- if  $\delta < \delta_c$ , then the extinction time is infinite with a positive probability that is,  $\mathbb{P}(\forall i, T_{stop}^i = \infty) > 0$ .

*Proof.* We first show that  $T_{stop}^i < +\infty$  almost surely for sufficiently large  $\delta$ . We observe that we can upper bound  $|C_t^i|$  (where  $|C_t^i|$  is the cardinal of  $C_t^i$ ) by  $Z_t$  almost surely for all  $t \geq 0$  where  $Z_0 = 1$  and  $(Z_t)_{t \geq 0}$  is a branching process. With a rate  $n(\beta^* - \beta_*)$  the transition from  $Z_t$  is from  $n$  to  $n + 1$  and with a rate  $n\beta_*$  this transition is from  $n$  to  $n - 1$ .

## NEURONAL NETWORK

We can therefore define for any bounded test function  $f$ , the associated infinitesimal generator of  $(Z_t)_{t \geq 0}$  as follows :

$$Af(n) = n[(\beta^* - \beta_*)(f(n+1) - f(n)) + \beta_*(f(n-1) - f(n))].$$

Take  $f(n) = n$ , we obtain :

$$Af = f[(\beta^* - \beta_*) - \beta_*] = f(\beta^* - \beta_*)(1 - \delta).$$

Then, for  $\delta > 1$ , we have  $Af(n) = -cf(n)$  where  $-c = (\beta^* - \beta_*)(1 - \delta)$ . Assuming  $x_t = \mathbb{E}(f(Z_t))$  and using the Itô formula, we have:

$$x_t = x_0 + \mathbb{E} \int_0^t Af(Z_s) ds = x_0 - c \int_0^t x_s ds = x_0 e^{-ct}.$$

Therefore, when  $t \rightarrow \infty$ , we have  $x_t \rightarrow 0$ . Which implies that if  $\delta > 1$ ,  $\mathbb{P}(T_{stop}^i < \infty) \geq \mathbb{P}(\lim_{t \rightarrow \infty} Z_t = 0) = 1$  thus ensuring that  $\delta_c \leq 1$ .

*Remark 4.8.*  $\mathbb{P}(T_{stop}^i < +\infty)$  is independent of neuron  $i$ . Therefore, if there exists a neuron  $i$  such that  $\mathbb{P}(T_{stop}^i < +\infty) < 1$  then for all  $i$  we have  $\mathbb{P}(T_{stop}^i < +\infty) < 1$ .

We now show that for all  $\delta < \delta_c$ ,  $T_{stop}^i = +\infty$  with positive probability.

For this proof, we will use the classical graphical construction of  $C_t^i$  (see Ferrari et al. [7], Griffeath [13]). We work within the space-time diagram  $\mathbb{Z} \times [0, \infty[$ . For each  $i \in \mathbb{Z}$ , we consider  $N_t^{i,s}$  and  $N_t^{i,p}$  two independent Poisson processes with respective intensities  $\beta_*$  and  $\beta^* - \beta_*$ . The jump times of  $N_t^{i,s}$  and  $N_t^{i,p}$  are respectively  $T_n^{i,s}$  and  $T_n^{i,p}$  for the neuron  $i$  after  $n$  jumps.

For each  $i \in \mathbb{Z}$ , we draw graphical sequences as follows. First draw arrows pointing from  $(i-1, T_n^{i,p})$  to  $(i, T_n^{i,p})$  and from  $(i+1, T_n^{i,p})$  to  $(i, T_n^{i,p})$  for all  $n \geq 1$ ,  $i \in \mathbb{Z}$ . Second,  $\delta$ 's at all  $(i, T_n^{i,s})$ , for all  $n \geq 1$ ,  $i \in \mathbb{Z}$ . We also suppose that time is going up which implies that we thus obtain a random graph  $\mathcal{P}$ . Let us say that there is a chain of vertical upward and horizontal directed edges in the random graph that leads from  $(i, 0)$  to  $(j, t)$  (with  $j \in \{i+1, i-1\}$ ) without passing through a  $\delta$ . Notice that  $C_t^i$  is the set of the clan of ancestors of site  $(i, t)$ , that is

$$C_t^i = \{j : \text{there is a path from } (i, 0) \text{ to } (j, t) \text{ for } j = i \pm 1\}.$$

It is obvious to notice that  $T_{stop}^i < \infty$  if and only if  $\bar{C}^i = \cup_{t \geq 0} C_t^i$  is a finite set. We will therefore show that  $\mathbb{P}(T_{stop}^i < \infty) = \mathbb{P}(|\bar{C}^i| < \infty) < 1$  for sufficiently small values of  $\delta$  using classical contour techniques. (see Griffeath [13].)

For this, on  $|\bar{C}^i| < \infty$ , we draw the contour of  $\bar{C}^i$  as follow.

Starting from  $(i - \frac{1}{2}, 0)$ . Let  $\Gamma$  be a possible path of the graph  $\mathcal{P}$ .  $\Gamma$  consists of  $4n$  alternating vertical and horizontal edges for some  $n \geq 1$  which we encode as a succession of direction vectors  $(D_1, \dots, D_{2n})$ . Each of the  $D_i$  can be one the seven triplets

$$dld, drd, dru, ulu, uru, urd, dlu,$$

where  $d, u, l$  and  $r$  stand for down, up, left and right, respectively. Note that  $uld$  cannot occur in a possible path  $\Gamma$  because the direction of  $uld$  is counter-clockwise. We start at  $(i - \frac{1}{2}, 0)$  and move clockwise around the curve.

The two figures below show examples of possible paths for  $n = 3$  and  $n = 4$ . Figure 1 shows a possible path with  $n = 3$  and in this case we have

$$\Gamma : ulu, ulu, urd, drd, drd, dlu.$$

For  $n = 4$ , Figure 4.2(b) gives

$$\Gamma : ulu, ulu, urd, drd, dru, urd, dld, dlu.$$

Writing  $N(dld), N(drd), \dots$  for the number of appearances of the different direction vectors, we have that  $N(dlu) = 1$  ( $dlu$  is the last triplet of which appears exactly one single time) and

$$N(dru) = N(urd) - 1 \leq n/2, \quad N(drd) + N(dru) + N(uru) + N(urd) = n.$$





Therefore, a very approximate upper bound on the total number of possible triplets  $(D_1, \dots, D_{2n})$  is given by  $4^{2n} = 16^n$ . We get for all  $\delta < \frac{1}{(16)^2}$ ,

$$\mathbb{P}(T_{stop}^i < \infty) \leq \delta + 4\delta^2 + \sum_{n \geq 3} (16)^n \delta^{n/2} = \delta + 4\delta^2 + \frac{(16\sqrt{\delta})^3}{1 - 16\sqrt{\delta}}.$$

We set  $\phi : \delta \mapsto \phi(\delta) = \delta + 4\delta^2 + \frac{(16\sqrt{\delta})^3}{1 - 16\sqrt{\delta}}$ . Then,  $\mathbb{P}(T_{stop}^i < \infty) \leq \phi(\delta)$ . As  $\delta \rightarrow 0$ ,  $\phi(\delta) \rightarrow 0$  which implies that there exists  $\delta_c$  such that  $\phi(\delta_c) = 1$ . As a consequence,  $\mathbb{P}(T_{stop}^i < \infty) < 1, \forall 0 < \delta < \delta_c$ .

We therefore conclude that  $\delta_c$  exists and  $0 < \delta_c \leq 1$ .  $\square$

*Remark 4.10.* Notice that,  $\delta \mapsto \mathbb{P}(T_{stop}^i < +\infty)$  is monotone (see [7, Lemma 5]). This implies the uniqueness of  $\delta_c$ .

#### 4.1. Some simulations

We simulate the state  $X_0(i)$  in the stationary regime for a fixed neuron  $i \in \mathbb{Z}$  at time 0 and estimate its density. The main purpose of this simulation is to have an idea about the theoretical distribution of  $X_0(i)$  in its stationary regime (see [9]) and whether this distribution is impacted by the specification of  $F^i$ .

We denote by  $\mathcal{D}$  the set of neurons which belong to a clan of ancestors of neuron  $i$  at a time  $t$  or to its neighborhood.

To do this, we apply the following algorithm:

##### Algorithm 4.11.

- (1) Initialize the family  $\mathcal{V}_{\rightarrow i}$  of non empty neighborhoods of the neuron  $i$
- (2) Initialize  $C_0^i = i$  the clan of ancestors of neuron  $i$  at time  $t = 0$ .
- (3) For all time  $t > 0$  we let  $C_t^i$  the clan of ancestors of neuron  $i$  at time  $t$
- (4) While  $|C_t^i| > 0$  (where  $|C_t^i|$  denotes the cardinality of  $C_t^i$ ) do
  - Determine the next jump time  $t_{next} > t$  in the clan of ancestors of neuron  $i$  at time  $t_{next}$  and in  $\partial_{ext}(\text{clan})$ , the correspondant neuron  $j$  and the nature of jump
  - If neuron  $j \in C_t^i$  and the jump is sure, then  $C_{t_{next}}^i = C_t^i \setminus \{i\}$
  - If  $j \in C_t^i$  and the jump is possible  $C_{t_{next}}^i = C_t^i$
  - If  $j \in V(C_t^i)$  (where  $V(C_t^i) := \cup_{j \in C_t^i} \mathcal{V}_{\rightarrow j}$ ) and the jump is sure, then  $C_{t_{next}}^i = C_t^i$
  - If  $j \in V(C_t^i)$  and the jump is possible  $C_{t_{next}}^i = C_t^i \cup \{j\}$
  - We update  $t \leftarrow t_{next}$

end While.
- (5) We determine the chronological list of the different jump times from 0 to the last time which makes the clan empty.
  - For each of these jump times, we indicate the associated neuron and the nature of the jump.
  - If the jump is sure, we simulate a random state following a distribution  $F^i$  at the neuron associated with this jump time.
- (6) We set  $m = \infty$ . While  $m > 0$  do
  - Let  $m$  be the rank of the last possible jump time  $T_m$  of  $\mathcal{D}$  in the chronology of jump times.

- Let  $k$  be the neuron associated with this jump.
- (7) We determine the rank  $r$  of the last sure jump time  $T_r > T_m$  of  $k$  in the chronology of jump times. The state of  $k$  is determined recursively from its state at time  $T_r$  to its state at time  $T_m$  as follows:
- For  $s \in \{1, \dots, r - m - 1\}$  let  $x =$  state of  $k$  at time  $T_{r-s+1}$ .
  - Let  $dt = T_{r-s+1} - T_{r-s}$  and  $j$  the neuron associated with the jump time  $T_{r-s}$ . The state of  $k$  at time  $T_{r-s}$  is  $x_{-dt}^k(x) + W_{j \rightarrow k} * \mathbf{1}_{\{\text{sure jump of } j \text{ at } T_{r-s}\}}$  with  $W_{j \rightarrow k}$  the inhibition weight of  $j$  on  $k$ .
  - We determine rather the occurrence is effective or not of the jump of  $k$  at time  $T_m$  thanks to its state at time  $T_{m+1}$ .<sup>1</sup>
  - If the jump is effective, we simulate a random state for  $k$  at time  $T_m$  following a distribution  $F^i$ . Otherwise, we determine the state of  $k$  at time  $T_m$  as  $x_{T_m - T_{m+1}}(x)$  where  $x =$  state of  $k$  at time  $T_{m+1}$ . Let  $m$  be the new rank of the last possible jump time of  $\mathcal{D}$  and repeat the procedure.

end While.

*Remark 4.12.* After this step, we know the exact nature of all jumps.

- (8) Determine for neuron  $i$  its first sure jump time  $T_n$  where  $n$  is the rank of this time in the chronology of jump times.
- (9) The state of neuron  $i$  is determined recursively from its state at time  $T_n$  to  $T_0$  as follows:
- For  $s \in \{1, \dots, n - 1\}$  let  $x =$  state of neuron  $i$  at time  $T_{n-s+1}$ .
  - Let  $dt = T_{n-s+1} - T_{n-s}$  and  $j$  the neuron associated with the jump time  $T_{n-s}$ . The state of neuron  $i$  at time  $T_{n-s}$  is  $x_{-dt}^i(x) + W_{j \rightarrow i} * \mathbf{1}_{\{\text{sure jump of } j \text{ at } T_{n-s}\}}$  with  $W_{j \rightarrow i}$  the inhibition weight of  $j$  on  $i$ .

*Remark 4.13.* The last value determined is the stationary state of neuron  $i$ .

*Remark 4.14.* This algorithm is inspired by [9, p. 20–21] which shows that if we find the state of a fixed neuron  $i \in \mathbb{Z}$  at time 0, it is necessarily the state of the neuron  $i$  in its stationary regime. The algorithm is not a proof in itself, but allows to have an idea of the theoretical distribution of the value of neuron  $i$  at time 0 in its stationary regime.

In the three following examples we consider  $\alpha_i(x) = x$ ,  $\beta_i(x) = 3 + \mathbf{1}_{\{x \leq 2\}}$ ,  $W_{i \rightarrow j} = 1$ . To verify if the distribution of inhibition state depends on the distribution  $F^i$ , we consider three different distributions for  $F^i$  that are  $\mathcal{E}(1)$ ,  $\mathcal{E}(10)$  and  $0.5\delta_1 + 0.5\delta_2$ . We simulate, with the algorithm described above  $N = 1000$  values for the inhibition state. We then estimate non parametrically the distribution of the inhibition state in these three cases of distribution  $F^i$  and we compare them.

The stationary distribution of the process in the three following cases seems to be continuous. We do not provide a proof here, this is outside the scope of this paper.

We can remark that the distribution of state of inhibition  $X_0(i)$  in stationary regime is concentrated in the interval  $(0, 4)$  when  $F^i = \mathcal{E}(1)$  whereas this distribution is rather concentrated on the interval  $(20000, 40000)$  when  $F^i = \mathcal{E}(10)$ . This shows that these two distributions of state  $X_0(i)$  are different.

<sup>1</sup>The jump occurs with a Bernoulli distribution with parameter  $(\beta(x_{-dt}^k(x)) - \beta_*) / (\beta^* - \beta_*)$

## NEURONAL NETWORK

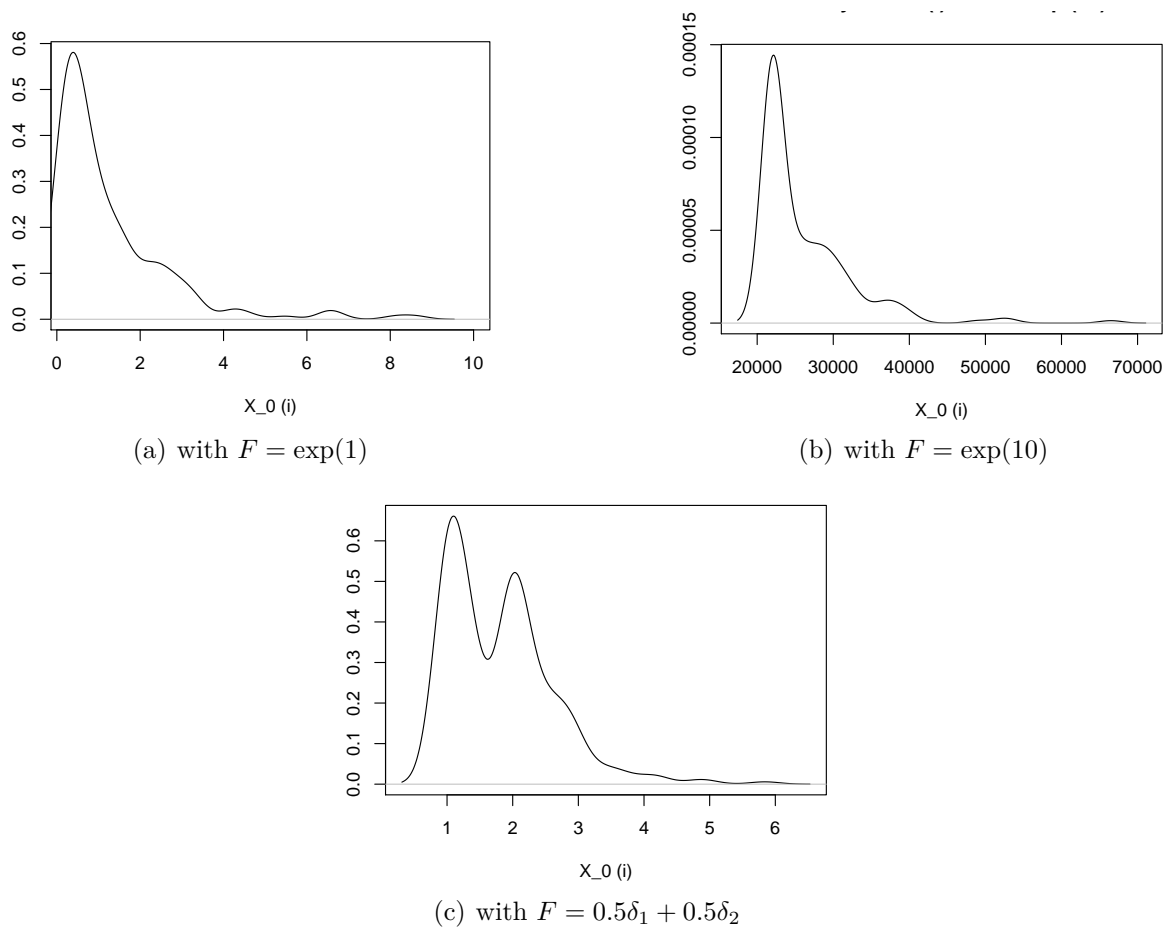


FIGURE 4.3. Densities of  $X_0(i)$

In this example, the distribution of the state of inhibition  $X_0(i)$  in stationary regime seems to be continuous although  $F^i$  is discrete. We do not provide a proof here, this is outside the scope of this paper. We observe two local extrema at 1 and 2 which are linked to the jumps because of the Dirac. These extrema suggest that jumps are very frequent in this process.

### Acknowledgements

The author thanks Eva Löcherbach for the many comments and discussions that led to this version of the paper, as well as the two referees who through their comments helped me to improve this paper.

## References

- [1] Morgan André. A result of metastability for an infinite system of spiking neurons. *J. Stat. Phys.*, 177(5):984–1008, 2019.
- [2] Krishna B. Athreya and Peter E. Ney. *Branching processes*. Courier Corporation, 1972.
- [3] Michel Benaïm, Stéphane Le Borgne, Florent Malrieu, and Pierre-André Zitt. Qualitative properties of certain piecewise deterministic MARKOV processes. *Ann. Inst. Henri Poincaré, Probab. Stat.*, 51(3):1040–1075, 2015.
- [4] Francis Comets, Roberto Fernández, and Pablo A. Ferrari. Processes with long memory: regenerative construction and perfect simulation. *Ann. Appl. Probab.*, 12(3):921–943, 2002.
- [5] Quentin Cormier, Etienne Tanré, and Romain Veltz. Long time behavior of a mean-field model of interacting neurons. *Stochastic Processes Appl.*, 130(5):2553–2595, 2020.
- [6] Marie Cottrell. Mathematical analysis of a neural network with inhibitory coupling. *Stochastic Processes Appl.*, 40(1):103–126, 1992.
- [7] Pablo A. Ferrari, Antonio Galves, Ilie Grigorescu, and Eva Löcherbach. Phase transition for infinite systems of spiking neurons. *J. Stat. Phys.*, 172(6):1564–1575, 2018.
- [8] Christine Fricker, Philippe Robert, Ellen Saada, and Danielle Tibi. Analysis of a network model. In *Cellular Automata and Cooperative Systems*, pages 159–172. Springer, 1993.
- [9] Antonio Galves, Nancy Lopes Garcia, Eva Löcherbach, and Enza Orlandi. Kalikow-type decomposition for multicolor infinite range particle systems. *Ann. Appl. Probab.*, 23(4):1629–1659, 2013.
- [10] Antonio Galves and Eva Löcherbach. Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets. *J. Stat. Phys.*, 151(5):896–921, 2013.
- [11] Antonio Galves, Eva Löcherbach, and Enza Orlandi. Perfect simulation of infinite range GIBBS measures and coupling with their finite range approximations. *J. Stat. Phys.*, 138(1):476–495, 2010.
- [12] Branda Goncalves, Thierry Huillet, and Eva Löcherbach. On decay-surge population models. <https://arxiv.org/abs/2012.00716>, 2020.
- [13] David Griffeath. The basic contact processes. *Stochastic Processes Appl.*, 11(2):151–185, 1981.
- [14] Sean P Meyn and Richard L Tweedie. Stability of Markovian processes III: Foster–Lyapunov criteria for continuous-time processes. *Adv. Appl. Probab.*, 25(3):518–548, 1993.
- [15] Philippe Robert and Jonathan Touboul. On the dynamics of random neuronal networks. *J. Stat. Phys.*, 165(3):545–584, 2016.