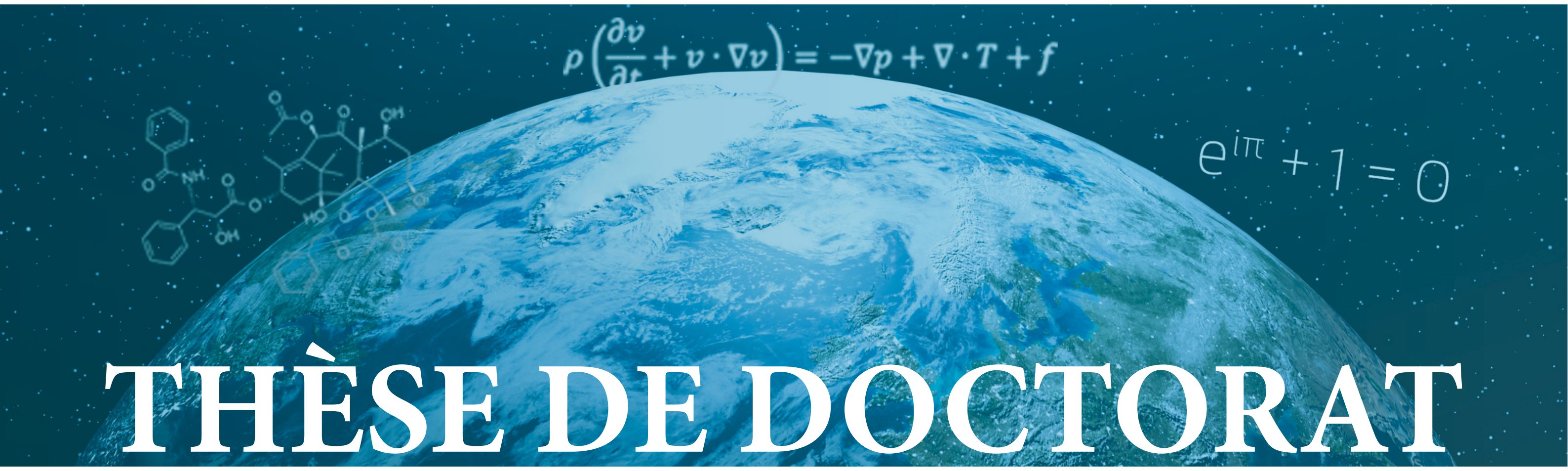
Kalikow decomposition for counting processes with stochastic intensity

Tien Cuong Phi

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE SCIENCES FONDAMENTALES ET APPLIQUÉES



Décomposition de Kalikow pour des processus de comptage à intensité stochastique

Tien Cuong PHI

Laboratoire J.A. Dieudonné

Présentée en vue de l'obtention du grade de docteur en mathématique **d'**Université Côte d'Azur
Dirigée par : Patricia Reynaud-Bouret
et co-dirigée par : Eva Löcherbach
Soutenue le : 14/06/2022

Devant le jury composé de

Patricia REYNAUD-BOURET	Supervisor	DR CNRS, LJAD et NeuroMod, Université Côte d'Azur
Eva LÖCHERBACH	Co-supervisor	Professor, Université Paris 1
Philippe ROBERT	Reviewer	DR INRIA, Inria de Paris
Nancy Lopes GARCIA	Reviewer	Professor, University of Campinas, Brazil
Etienne TANRÉ	Examinator	CR INRIA, Inria Centre d'Université Côte d'Azur
Alexandre MUZY	Examinator	CNRS, I3S, Inria Centre d'Université Côte d'Azur

Décomposition de Kalikow pour des processus de comptage à intensité stochastique

Kalikow decomposition for counting processes with stochastic intensity

Membres du jury:

Rapporteurs

Rapporteur	Philippe Robert	DR INRIA	Inria de Paris
Rapportrice	Nancy Lopes Garcia	Professeure	Université de Campinas, Brésil
Examinateur	Alexandre Muzy	CNRS	I3S, Inria Centre d'Université Côte d'Azur
Examinateur	Etienne Tanré	CR INRIA	Inria Centre d'Université Côte d'Azur
Directrice	Patricia Reynaud-Bouret	DR CNRS	LJAD et NeuroMod, Université Côte d'Azur
Co-directrice	Eva Löcherbach	Professeure	SAMM, Université Paris 1

Abstract

The goal of this thesis is to construct algorithms which are able to simulate the activity of a neural network. The activity of the neural network can be modeled by the spike train of each neuron, which are represented by a multivariate point processes. Most of the known approaches to simulate point processes encounter difficulties when the underlying network is large.

In this thesis, we propose new algorithms using a new type of Kalikow decomposition. In particular, we present an algorithm to simulate the behavior of one neuron embedded in an infinite neural network without simulating the whole network. We focus on mathematically proving that our algorithm returns the right point processes and on studying its stopping condition. Then, a constructive proof shows that this new decomposition holds for on various point processes.

Finally, we propose algorithms, that can be parallelized and that enables us to simulate a hundred of thousand neurons in a complete interaction graph, on a laptop computer. Most notably, the complexity of this algorithm seems linear with respect to the number of neurons on simulation.

Keywords: stochastic simulation, perfect simulation, Kalikow decomposition, Hawkes process, thinning algorithm, large-scale simulation, point process, stochastic process.

Résumé

L'objectif de cette thèse est de construire des algorithmes capables de simuler l'activité d'un réseau de neurones. L'activité du réseau de neurones peut être modélisée par le train de spikes de chaque neurone, qui sont représentés par un processus ponctuel multivarié. La plupart des approches connues pour simuler des processus ponctuels rencontrent des difficultés lorsque le réseau sous-jacent est de grande taille.

Dans cette thèse, nous proposons de nouveaux algorithmes utilisant un nouveau type de décomposition de Kalikow. En particulier, nous présentons un algorithme permettant de simuler le comportement d'un neurone intégré dans un réseau neuronal infini sans simuler l'ensemble du réseau. Nous nous concentrons sur la preuve mathématique que notre algorithme renvoie les bons processus ponctuels et sur l'étude de sa condition d'arrêt. Ensuite, une preuve constructive montre que cette nouvelle décomposition est valable pour divers processus ponctuels.

Enfin, nous proposons des algorithmes, qui peuvent être parallélisés et qui permettent de simuler une centaine de milliers de neurones dans un graphe d'interaction complet, sur un ordinateur portable. Plus particulièrement, la complexité de cet algorithme semble linéaire par rapport au nombre de neurones à simuler.

Mots clef: simulation stochastique, simulation parfaite, décomposition de Kalikow, processus de Hawkes, algorithme de rejet, simulation à grande échelle, processus ponctuel, processus stochastique.

Acknowledgements

When I write these lines, I know that it marks the end of a long and memorable journey in my life . I am grateful to many people for the help and support during the various stage of writing my PhD thesis. First and foremost, I would like to thank you the defense committee: Philippe Robert, Nancy Lopes Garcia, Etienne Tanre, and Alexandre Muzy for crucial feedbacks which greatly improve my thesis. Manon and Roland played a crucial role in my committee de these over 3 years, providing endless assistance and for this, I am grateful to them.

Next and certainly, most importantly, I turn to Patricia and Eva. Surely a page, even a book, is not enough for me to express my gratitude to them. No words are enough to express my gratitude. I always consider them as my second mothers. They give me advice, both in math and in real life. I have known Patricia for 5 years. Perhaps no one understands my abilities better than Patricia. Patricia taught me so much and spent a lot of time talking to me in her limited time. In Patricia, I found constant creativity, enthusiasm for science, and daring to take on challenges. She also taught me lessons about interaction with other colleagues. Although I don't have much time to work directly with Eva, in return, every time I get to meet Eva, she always takes precious time to talk to me. Her extensive knowledge has helped me a lot to get to where I am today.

I would like to thank the friends who have helped me a lot along this way. First, I would like to express my sincere thanks to Gilles, who has been very supportive over the years, both in work and real life. Next, I would like to thank Gaetan, my best friend over the years. An enthusiastic and funny friend. Gaetan's help certainly got me through the most difficult first year in France. I also thank Paul, who has contributed significantly to helping me complete the final chapter of this thesis. Paul taught me a lot about programming and suggested me lots of good questions. I would also like to thank my friends: Cyrille, Giulia, Julien, Stephano, Marie, etc.

I also do not forget to thank my Vietnamese friends, such as Hai, Tuan, and Lam. Thank you Bien Thoa, Trang Tu, and most especially Quang Huong, we have given each other invaluable care and help. Thank you, Phuong Anh, for editing this English manuscript.

I am forever indebted to my parents, and my siblings, who have always stand by my side and supported me. Most especially, I would like to thank to my wonderful life partner Ngan and my son Bach, who has greatly helped me to complete this journey. I hope that through those efforts, I instill in him the belief that hard work does pay off.

I would like to thank my colleagues in the LJAD lab, the secretaries of the team in the laboratory like Narymen, etc.

I know that there are many other friends, who for a moment I cannot remember. I would like to thank all of you.

List of Figures

Contents

1 Introduction

1.1 Mathematical background

The main mathematical object underlying this thesis is the temporal point process. Let us start with some notation and definitions.

1.1.1 Point process

A point process Z in $\mathbb{R}$ can be viewed as a (countable) collection of times in $\mathbb{R}$, denoted by $(T_n)_{n\in\mathbb{Z}}$, with the convention that $T_0 \leq 0 < T_1$. We will call $\mathscr{B}(\mathbb{R})$ the Borel subsets of $\mathbb{R}$. For any bounded measurable set A belonging to $\mathscr{B}(\mathbb{R})$, we denote $Z(A)$ the cardinal of $Z \cap A$.
Throughout this thesis, we will only consider the simple point processes. A point process Z is simple if $\mathbb{P}(\forall t,\ Z([t]) = 0 \text{ or } 1) = 1$ with the convention that $[t] := [t, t]$.
The point process Z can also be defined by its associated discrete random measure $dZ_t := Z(dt)$, i.e the random measure on $(\mathbb{R}, \mathscr{B}(\mathbb{R}))$ such that, for any positive measurable function f, we have $\int_{\mathbb{R}} f(t)dZ_t = \sum_{k\in\mathbb{Z}} f(T_k)$. Associated to the point process Z, we also define a counting process $Z_t := Z((0, t])$ if $t \geq 0$ and $Z_t := -Z([t, 0))$ otherwise.
Moreover, associated to the process Z, we consider the canonical filtration $\mathscr{F}_t^Z$ defined by

$$\mathscr{F}_t^Z = \sigma(Z \cap (-\infty, t]).$$

If the process Z is such that, for any t

$$\mathscr{F}_t \supset \mathscr{F}_t^Z$$

then $\mathscr{F}_t$ is called a history of the process Z. We will always be in this case in this manuscript.

It is more convenient to study point processes via their stochastic intensity, which is defined in the next section.

1.1.2 Stochastic intensity

To facilitate the writing, in this section, we will only consider the point process Z in $\mathbb{R}_+$. The result can be easily extended in a more general framework.

The $\mathscr{F}$-predictable process $(\phi_t)_{t\geq 0}$ is called $\mathscr{F}$-intensity of Z if $Z_t - \int_0^t \phi_s ds$ is a $\mathscr{F}_t$ martingale. This definition opened a new approach to study point process with abundant tools of martingale theory [2].

In practice, to verify that ϕ_t is the $\mathscr{F}$ intensity of the process Z, we often use the following equivalent definition of the stochastic intensity (Definition D7 of [2]).

Definition. Let Z_t be a point process adapted to some history $\mathscr{F}_t$ and let ϕ_t be a nonnegative $\mathscr{F}_t$ predictable process such that for all $t \geq 0$

$$\int_0^t \phi_s ds < \infty \quad a.s$$

If for all nonegative $\mathscr{F}_t$ predictable C_t we have

$$\mathbb{E}\Big(\int_0^\infty C_s dZ_s\Big) = \mathbb{E}\Big(\int_0^\infty C_s \phi_s ds\Big).$$

Then we say Z_t admits the $\mathscr{F}_t$ (predictable) intensity ϕ_t.

Most of the models under consideration in this thesis come from neuroscience, that is the object of the next section.

1.2 Neuroscience motivation

Neurons or nerve cells are the fundamental units of the brain and nervous system. The human brain has about 10^{11} neurons that fire electrochemical signals to communicate with each other via specialized connections called *synapses*. These signals are transmitted from the cell body of a neuron through the *axon* (then synapse) to the next neuron (see Figure 1.1).

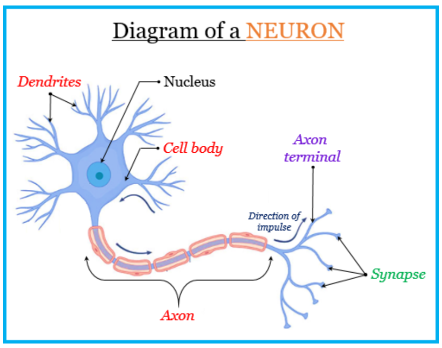


Figure 1.1: Neuron
Source: tutorix.com

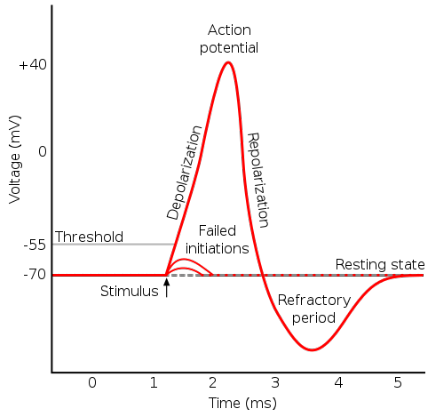


Figure 1.2: Action potential
Source: teachmephysiology.com

A neuron receives thousands of synaptic inputs from other neurons, then sums them up (*synaptic integration*), if this sum is large enough (see Figure 1.2), it produces an *action potential* also called spike. The action potentials of a given neuron having roughly all the the time the same shape, therefore, we are only interested by the time of their emission. One way to present the neural activity is to record the occurrence of spikes, which we call *spike train*. That is why we use point processes to model the neural activity and in particular we use Hawkes process, that is presented in the upcoming section. The average number of spikes over a time unit is called *firing rate*.

1.3 Hawkes process

Hawkes process is firstly introduced in the papers [12, 13]. It then receives a lot of interest in the last decade and has numerous applications in various fields, to name but a few:

- in seismology [24]
- in finance [1]
- in genome analysis [31]
- spread of infection disease [5] .

Throughout this thesis, the integral $\int_a^b$ stands for $\int_a^{b-}$ or $\int_{[a,b)}$ for the reason of predictability, if we do not mention it differently. Most of the time, we will work with multivariate and non-linear Hawkes process. Let us consider $\mathbf{I}$ the index set, $\mathbf{I} = \{1, 2, \ldots, N\}$. A multivariate Hawkes process $(Z^1, \ldots, Z^N)$ is defined by its stochastic intensity for $i = 1, \ldots, N$, which has the following form in general:

$$\phi_i(t|\mathscr{F}_t) = \psi_i\left(\sum_{j\in\mathbf{I}} \int_{-\infty}^{t} h_{ji}(t-s)dZ_s^j\right) \tag{1.3.1}$$

where ψ_i is called the associated intensity function and h_{ji} measures the influence of index j on index i.

In neuroscience, the intensity ϕ_i represents the firing rate of neuron i. Here, the firing rate of neuron i depends on the spikes of all the neurons before time t. This formula imitates very well the synaptic integration procedure of a neural network. In addition, for mathematical purposes, we often assume that ψ_i is continuous function (or even Lipschitz for the reason of stability [3, 4]). The continuity assumption here also ensures that the ϕ_i is predictable [32]. In addition, if we do not state otherwise, we always assume that h_{ji} is a L^1 function, i.e $\|h_{ji}\|_{L^1} = \int_0^\infty |h_{ji}(t)|dt < \infty$. We also suppose that ψ_i is locally integrable to avoid explosion [32].

In the following, we will present several type of Hawkes processes. We begin with linear Hawkes process, which is obtained by considering the function ψ_i in Equation 1.3.1 has the following form $\psi_i(x) := \mu_i + x$. Then the conditional intensity is given by

$$\phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} \int_{-\infty}^{t} h_{ji}(t-s)dZ_s^j \tag{1.3.2}$$

where we keep all conditions as stated earlier, μ_i is a positive number, which represents for the spontaneous rate of the neuron i and h_{ji} is positive.

In neuroscience, it is reasonable to do not take into account the spikes which appear far from the time of consideration. It leads us to consider the linear Hawkes process with bounded support of interaction A, with intensity

$$\phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} \int_{t-A}^{t} h_{ji}(t-s)dZ_s^j \tag{1.3.3}$$

where A is a given positive number.

As notice in the Figure 1.2, after a neuron emits a spike, the firing rate of this neuron rapidly decreases to zero. Then it follows by a refractory period, in which the neuron is unable to produce a new spike. This can be modelled by the nonlinear Hawkes process with hard refractory period [29] with intensity

$$\phi_i(t|\mathscr{F}_t) = \psi_i\left(\sum_{j\in \mathrm{I}} \int_{-\infty}^{t} h_{ji}(t-s)dZ_s^j\right)\mathbb{1}_{t-L_i(t)>\delta} \tag{1.3.4}$$

where δ is a positive number, representing the length of the refractory period and $L_i(t) = \sup\{T \in Z^i \text{ such that } T < t\}$ the last spiking time before time t.

In this thesis, to simplify, we only focus on the positive interaction function h_{ji}, means we only consider the excitatory effects of neurons. However, our results are possible extended to the case of excitatory and inhibitory also. We left it for future works.

In mathematics, Hawkes processes have been extensively studied in probability theory, statistics and computer science:

1. stationary and stability [3]
2. long time behavior [32]
3. mean field limits [4]
4. simulation [19, 20, 21]

We focus in particular on the simulation of Hawkes processes. Simulation helps us to predict the pattern of point process in the future, for model checking, see the further discussions in [30]. In particular, brain simulation is received a lot of attentions recently. There exists several grand projects such as the Human Brain Project in Europe, the Brain Mapping in Japan and the Brain Initiative in the United States. In the next section, we present simulation algorithms of point processes.

1.4 Simulation algorithms and Kalikow decomposition

1.4.1 Ogata's algorithm

We begin with the simulation algorithm proposed by Lewis and Shedler [18] in 1979, which is applied to nonhomogeneous Poisson processes. Inspired by this method, two years later, Ogata [23] introduced a thinning algorithm to simulate point processes having stochastic intensity. Throughout this thesis, we call classical Ogata's algorithm to refer to this method. In the following, we rewrite Proposition 1 of [23]. Say we want to simulate a multivariate point process $Z = (Z^i)_i$ with intensity $(\phi_i)_{i\in \mathrm{I}}$ in the time interval $[0, t_{max}]$ Moreover, we assume that there is no point before time 0, i.e $Z_0^i = 0, \forall i$. Then Proposition 1 of [23] can be rewritten as follow.

Proposition. There exist a $\mathscr{F}_t$ predictable process Λ_t which is defined path-wise such that

$$\sum_{i=1}^{N} \phi_i(t|\mathscr{F}_t) \le \Lambda_t$$

and we set

$$\phi_{N+1}(t|\mathcal{F}_t) = \Lambda_t - \sum_{i=1}^{N} \phi_i(t|\mathcal{F}_t).$$

Let us denote $T_1, T_2, \dots, T_{\Pi_{t_{max}}}$ be the points of process Π with the intensity Λ_t. We assign a mark $p = 1, \dots, N+1$ to point T_k if

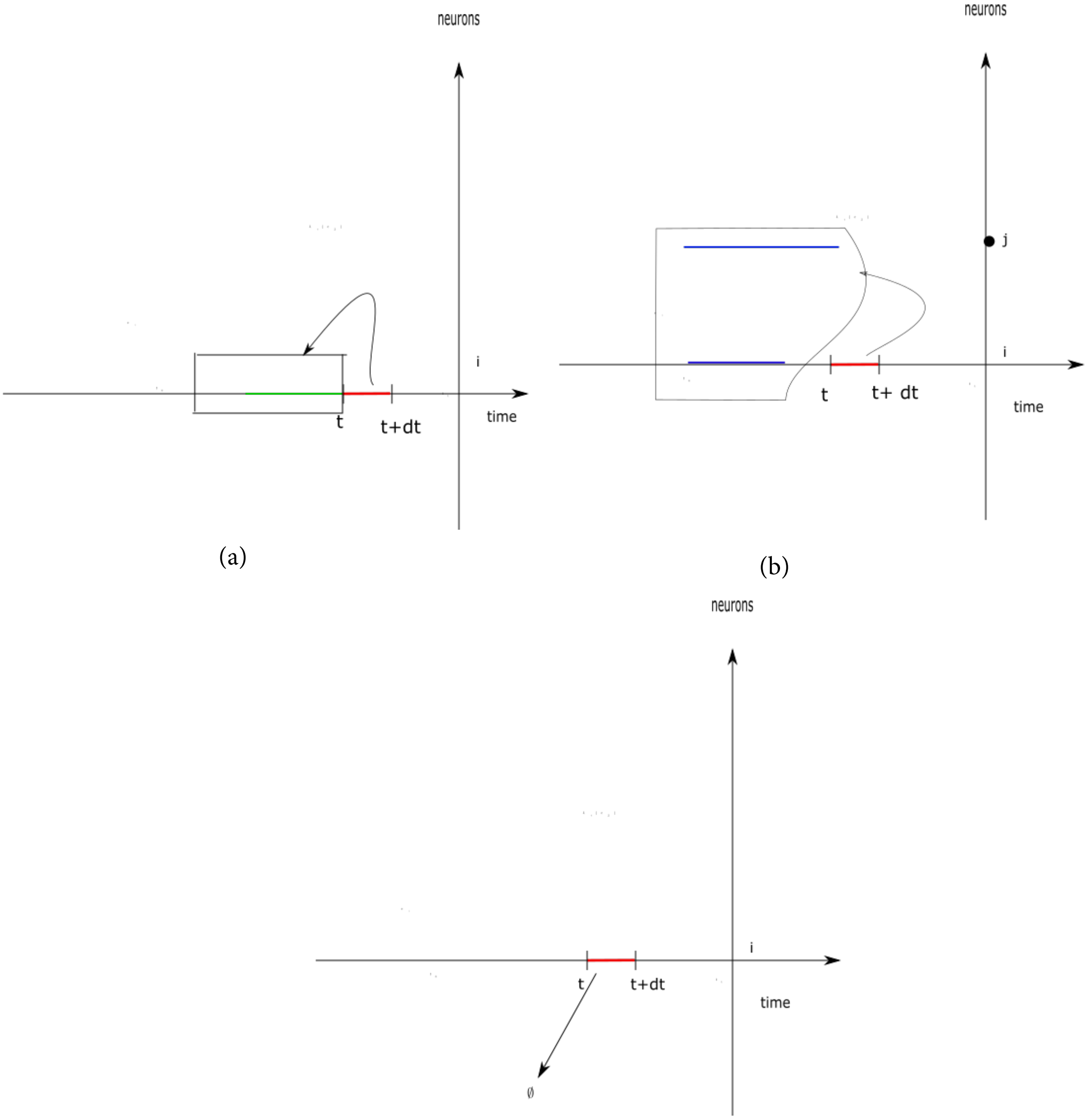
$$\sum_{i=1}^{p-1} \phi_i(T_k|\mathcal{F}_{T_k})/\Lambda_{T_k} < U_k \leq \sum_{i=1}^{p} \phi_i(T_k|\mathcal{F}_{T_k})/\Lambda_{T_k}$$

where U_k is an uniform random variable on $[0,1]$. Then the points with mark $p = 1, \dots, N$ forms a multivariate point process with intensity $(\phi_i(t|\mathcal{F}_t))_i$.

This method is quite generic, since it does not need any specific condition on the conditional intensity but it contains two main drawbacks. First, we can not obtain a stationary version of the Hawkes process since we start at time 0 without a point before 0. More importantly, at each point T_k, in order to assign a mark, we need to compute the cumulative sum of the intensities. However, this cumulative sum becomes extremely hard to compute when the number of neurons N is big. This is one of the main motivations of the thesis, keeping in mind that the neural network of the humain brain is of size $N = 10^{11}$. Recently, there exists another approach by Mascart et al. [19], in which, they also aim to simulate such a huge neural network. Their algorithm is a variant of Ogata's algorithm. However, their network needs to be sparse, in the sense that one neuron is only connected to very few other neurons. In our thesis, we can consider a neural network which is complete, i.e each neuron connects to all the other neurons and we can even consider the infinite networks. In the next section, we present how we are able to do so.

1.4.2 Kalikow decomposition

This section is devoted to introducing the Kalikow decomposition to improve the classical Ogata thinning algorithm. To begin, we define what a neighborhood is. Intuitively, the neighborhood of (i,t) is a part of the past that is needed to compute the stochastic intensity $\phi_i(t|\mathcal{F}_t)$. The rigorous mathematical definition can be found in Chapter 2 and 3.



(a) (b) (c)

Figure 1.3: A neighborhood

In Figure 1.3, we present three different possibilities of a space-time neighborhood. Note that, in an informal way, the intensity function $\phi_i(t|\mathscr{F}_t)$ represents for the probability obtaining a new point in a small interval $[t, t+dt]$ given the past before time t. To decide whether we have a new point in $[t, t+dt]$, we need to look at either a neuron and a fraction of time before t (the green line in Figure 1.3(a)), two neurons and two different fractions of times before t (the blue lines in Figure 1.3(b)) or empty set (Figure 1.3(c)), i.e independent to the past.

The whole point of Kalikow decomposition is to pick at random a neighborhood in a family of space-time neighborhoods (neighborhood family) according to a certain distribution. This can be done thanks to the Kalikow decomposition, which is presented in the following.

We say that a stochastic intensity $\phi_i(t|\mathscr{F}_t)$ admits a Kalikow decomposition with respect to the neighborhood family $\mathbf{V}$ if there is a sequence of function $\phi_i^{v_k}(t|\mathscr{F}_t)$ which only depends on the information in the neighborhood v_k and the probability distribution on $\mathbf{V}$, $\lambda_{i,t}(v)$ such that

$$\phi_i(t|\mathscr{F}_t) = \lambda_{i,t}(\emptyset)\phi_i^{\emptyset}(t) + \sum_k \lambda_{i,t}(v_k)\phi_i^{v_k}(t|\mathscr{F}_t)$$

with $\lambda_{i,t}(\emptyset) + \sum_k \lambda_{i,t}(v_k) = 1$.
The Kalikow decomposition can be interpreted as follows. The intensity function $\phi_i(t|\mathscr{F}_t)$ in some cases might depend on the whole past. Instead of taking the whole information in the past, we draw a random neighborhood V, that is a finite part of the past according to the distribution $\lambda_{i,t}(.)$. Once the neighborhood is chosen, assume that is is v_k, we then compute the intensity function according to $\phi_i^{v_k}(t|\mathscr{F}_t)$ which depends only on the points in v_k. The information in v_k is much less than the whole past, which hopefully helps us to compute the intensity more efficiently.

Kalikow decomposition [16] was largely studied in the context of stochastic processes having long memory [6, 10, 11, 25]. But the decomposition was applied to transition probability rather than to a stochastic intensity in our study. As a result, they consider only discrete-time processes. Despite numerous studies on discrete-time processes, up to our knowledge, there exists only one work on continuous processes by Hodara and Löcherbach [14]. It is worth noting that, in their paper, the intensities are assumed to be bounded by a constant, and $\lambda_{i,t}(v_k)$ are not deterministic but depends on the realization of a dominating Poisson process, which in practice need to be simulated at prior to write the Kalikow decomposition. To distinguish, we call such decomposition, the conditional Kalikow decomposition and the one defined with deterministic λ by unconditional Kalikow decomposition. A very natural question arises, does this unconditional Kalikow decomposition exists in different examples? If it is the case, how to write it? How we can apply this decomposition in the simulation?

Remark 1. It is worth noting that, at this stage the distribution function $\lambda(.)$ in Kalikow decomposition can be either deterministic or random. We have not proved yet such a deterministic Kalikow decomposition exists in the continuous time process.

1.4.3 Perfect simulation

In this section, we investigate several perfect simulation algorithms. By perfect simulation, we mean it can simulate the stationary distribution of a finite number of components in a potentially infinite network within a given finite space-time window. To explain clearly and coherently, we first examine the algorithms for discrete-time processes then continuous-time processes. This thesis mostly deals with Hawkes processes, which are in particular a process with long memory. Therefore, we will focus to the perfect algorithm developed by Comets et al. [6] in the discrete-time processes and compare it with the famous perfect simulation, named Coupling From the Past (CFTP) of Propp and Wilson [28]. Some other approaches to perfect simulations can be found at [8].

Perfect simulation is a widely discussed topic since the first seminal paper of Propp and Wilson [28]. The goal of this paper is to simulate the invariant measure of Markov chains. Following Propp and Wilson [28], fixed a time 0, for a fixed predeterminded time in the past, we simulate one trajectory per initial state. If the trajectories coalesce, the common value at time 0 is picked under the stationary distribution. If the trajectories do not coalesce, we extend their paths further in the past. Theorem 1 of [28] showed that if we start from a far enough time in the past, the trajectories starting from all possible initial states coalesce.

Different from the Propp and Wilson's method, perfect simulation in the paper of Comets et al. [6] used to simulate the invariant measure of the processes with long memory, which is more general than Markov chains and Markov processes. This method is based on the regenerative construction of processes, which is (as the authors cited) introduced in the paper [9]. In their method, at each instant n, there is a (random) number $k_n \geq 0$ such that the distribution of the move $n + 1$ is the same for all histories agreeing the k_n preceding instants. This independence from k_n remote past yields to the times τ such that preceding past before τ are irrelevant for future moves. If there exist $\tau > -\infty$ such that $\tau \leq n - k_n$ for all $n \geq \tau$, we call such τ the regeneration time. Moreover in Section 2 of [6], the authors presented explicitly the regenerative construction.

This second approach by Comets et al. relies on the regenerative representation of transition probability. Recently, by using the Kalikow decomposition's notation, Galves et al. provided a more general approach [10, 11] with the same spirit of Comets's approach. In short, as discussed in the previous section, Kalikow decomposition allows us to (randomly) determine a space-time neighborhood, which then constitute a clan of ancestors of the interested point. Ancestor of a point T is the point that might influence to T (see Chapter 3 to more rigorous definition). By applying the Kalikow decomposition to these obtaining points in the clan of ancestor, we obtain the ancestors of ancestor of the interested point. Repeat this procedure until all the points either having empty neighborhood or visited neighborhood, means it only contains the visited points. By doing so, we create a branching processes containing all influent points to the interest point. Denote τ the point with minimum in time. We see that all history before time τ is irrelevant to determine the state of the interested point and that this property is similar to [6]. Back to our perfect simulation algorithm, once the clan is completely constructed, we process with the Forward algorithm. In which, we begin with the smallest point. Then we continue with the point whose neighborhood is completely determined. We repeat this until the state of interested point is determined.

Most the existing papers focus on discrete time processes, in this thesis, we want to provide Perfect simulation for continuous time processes. Therefore, the approach of Comets need some adaptation. Indeed, we will need to decompose conditional intensities rather than transition probabilities. This leads to serious difficulties that usually prevent a more practical application of the Perfect Simulation algorithm. Again, up to our knowledge, the only work dealing with continuous time counting processes, is the one by Hodara and Löcherbach [15]. Their decomposition is constructed under the assumption that there is a dominating Poisson process on each of the nodes, from which the points of the processes under interest can be thinned by rejection sampling (see also [22] for another use of thinning in simulation of counting processes). To prove the existence of a Kalikow decomposition and go back to a more classic discrete time setting, the authors need to freeze the dominating Poisson process, leading to a mixture, in the Kalikow decomposition, that depends on the realization of the dominating Poisson process. Such a mixture is not accessible in practice, and this prevents the use of their Perfect Simulation algorithm for more concrete purposes than mere existence.

There also exists other approaches to do perfect simulation for continuous point processes:

- by Møller and Rasmussen [21]: this method only applies for linear Hawkes processes which can be presented as Poisson cluster processes. Then the length distribution of the clusters are required to perform simulation in their algorithm. However as they discussed, these are not known even in simple cases. Therefore, the authors used the idea of dominated Coupling From The Past ([17]), they replaced the (unkown) cumulative distribution function by the upper and lower bound of it. By this method, they create coupled upper and lower process. These processes will converge to the true Hakwes processes.

- by Dassio and Zhao [7]: this method exploit some special properties of exponential Hawkes process.

1.5 Presentation of the thesis

In Chapter 2, we introduce a new simulation algorithm in the stationary (time-homogeneous) system using the unconditional Kalikow decomposition. We apply this algorithm to a neuroscience question. More precisely, we study the behavior of one part of a neural network with potentially an infinite number of neurons. We consider $\mathbf{I}$ the set of neurons, which might be infinite (countable). Throughout Chapter 2, we focus on the following conditional intensity

$$\phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} w_{ij}(Z^j([t-A,t)) \wedge M)$$

where M is a given positive number and $w_{ij} \in [0,1]$ is the proportion to the interaction between neuron i and j such that $\sum_j w_{ij} < 1$ for all i. Clearly, within this formula, the conditional intensity is bounded. We can apply Ogata's thinning algorithm [23]. However, to do so, we need to simulate the whole system in order to simulate one neuron. In addition, starting from time 0, it does not go backward in time, therefore we can not simulate a Hawkes process in a stationary regime. There exist a work of Moller and Rassmussen [21] that provides a perfect simulation but needs the branching structure of the process to do so. In this chapter, we present another approach that also overcomes this flaw. We define a new type of Kalikow decomposition in continuous time (Definition 2.4.1 of Chapter 2).

Definition. We say that the process admits *a Kalikow decomposition* with bound M and neighborhood family $\mathbf{V}$, if for any neuron $i \in \mathbf{I}$, for all $v \in \mathbf{V}$ there exists a non negative M-bounded quantity $\phi^v_{i,t}$ whose value only depends on the points appearing in the neighborhood v, and a probability density function $\lambda_i(.)$ such that

$$\phi_i(t \mid \mathscr{F}_t) = \lambda_i(\emptyset)\phi_i^{\emptyset}(t) + \sum_{v\in\mathbf{V}, v\neq\emptyset} \lambda_i(v) \times \phi_i^v(t|\mathscr{F}_t) \tag{1.5.1}$$

with $\lambda_i(\emptyset) + \sum_{v\in\mathbf{V}, v\neq\emptyset} \lambda_i(v) = 1$.

Remark 2. Note that, at this stage, we only define the Kalikow decomposition for a bounded intensity and also we need to restrict our self to bounded ϕ^v.

Within this new type of Kalikow decomposition, we propose a theoretical algorithm that combines the idea of Ogata algorithm and Kalikow decomposition, see Algorithm 2 of Chapter 2. But this algorithm is purely theoretical since the computation of $\phi_i^{V_T}$ depends on the points in V_T, which are not known at this stage. To deal with this problem, we propose a modified version of the Perfect simulation, which we call Backward Forward algorithm (Algorithm 3). We prove that this algorithm ends in finite time almost surely in Proposition 1. Moreover, this algorithm also returns the right intensity process, which is proved in Proposition 2.

In Chapter 3, we show that unconditional Kalikow decomposition exists in various types of Hawkes processes. In this chapter, we also define for the first time an unconditional Kalikow decomposition in general case (Definition 3.3.3).

Definition. We say a process Z^i for some $i \in \mathbf{I}$ admits the Kalikow decomposition with respect to a neighborhood family $\mathbf{V}$ if for all t, for any $v \in \mathbf{V}$ there exists a cylindrical function $\phi_i^v(t|.)$ taking values in $\mathbb{R}^+$, whose value only depends on the points appearing in the neighborhood v, and a probability density function $\lambda_i(.)$ such that

$$\phi_i(t|\mathscr{F}_t) = \lambda_i(\emptyset)\phi_i^{\emptyset} + \sum_{v\in\mathbf{V}, v\neq\emptyset} \lambda_i(v)\phi_i^v(t|\mathscr{F}_t) \tag{1.5.2}$$

with $\lambda_i(\emptyset) + \sum_{v\in\mathbf{V}, v\neq\emptyset} \lambda_i(v) = 1$.

Remark 3. Note that, in this definition of Kalikow decomposition, there is no restriction to ϕ^v and the probability distribution λ is deterministic.

Next, very naturally, we show how to obtain the unconditional Kalikow decomposition for a various processes. This is a very constructive method and it also proved the existence of unconditional Kalikow decomposition, see Proposition 3 of Chapter 3.

On the other hand, for the nonlinear Hawkes process, in particular, we also propose another method using Taylor expansion in Proposition 5. We extend the definition of the random neighborhood in Chapter 2 and introduce the modified Perfect simulation with the general random neighborhood, see Backward-Foward algorithm in Section 3.5 of Chapter 3. Similar to the previous chapter, Proposition 7 gives the stopping condition for the Backward-Foward algorithm and Proposition 6 shows that it provides a right intensity process. In this chapter, we also study the complexity of the algorithm in terms of the number of simulated points (see Proposition 8 of Chapter 3). Based on these discussions, we show how to optimize the Kalikow decomposition in the last section of this chapter.

In Chapter 4, based on the theory developed in Chapter 3, we present several new simulation algorithms for point processes. We focus on two different hypotheses: either the conditional intensity is bounded by a deterministic number (Definition 7) or it is bounded by a predictable function (Definition 8).

In the first case, we rewrite Ogata's algorithm in our settings (see Algorithm 4). This version is sequential. Notably, we construct a parallelized version of Ogata's algorithm (see Algorithm 5). Next, by adding Kalikow decomposition, we propose two new algorithms, one is purely sequential (Algorithm 6), the other is partially parallelized (Algorithm 7).

In the second case, when the intensity is bounded by a predictable function, we construct two sequential algorithms: one we call Ogata's algorithm (Algorithm 8), the second one is KalikowOgata algorithm (Algorithm 9).

In the deterministic bounded case, we design an algorithm for a particular situation, namely the Hawkes process with a refractory period (Section 4.3.5), whereas in the second case, we focus on proving mathematical the correctness of the algorithms (Section 4.4.3). In the last section, we provide several statistical tests and execution times of algorithms in Section 4.5. Notably, the Kalikow Ogata algorithm is showed to be significantly better than the classical Ogata algorithm and to have linear complexity even for a complete graph on simulation.

We left chapters 2, 3, 4 under their article form except for a few modifications in order to preserve consistency between the chapters. In particular, some definitions may be redundant with this introduction.

1.5.1 Main contributions

First and foremost, my work is mostly inspired by practical questions. Therefore, I rather focus on the possibility of implementing the algorithms than the theoretical question about the existence of invariant measures like in [6, 10, 11, 14]. For example, in the paper [14] (which is closest to our work and is also the only work in continuous time processes), the authors need to simulate the whole underlying network to obtain Kalikow decomposition and then do a perfect simulation. From a computational point of view, this approach is unfeasible (both for finite and infinite network). Our approach only simulates parts of the dominating Poisson processes which are needed (in the bounded case). More importantly, as we discussed earlier in Section 1.4.2, going from conditional Kalikow decomposition to the unconditional one is crucial in practice. In addition, many numerical results are presented in Chapter 2 and Chapter 4 together with numerous discussions.

Besides, this thesis also has some theoretical contributions. The previous works on Kalikow decomposition [6, 9, 10] mostly relied on the continuity assumption, which is not necessary in the thesis. The Linear Hawkes process which is indeed not satisfied the continuity assumption with the "nested" neighborhood, therefore the classical Kalikow decomposition can not be obtained. However, by considering the Linear Hawkes process with specific choice of neighborhood, we obtain Kalikow decomposition, see Chapter 3. Lastly, in Chapter 4 of the thesis, we also proposed an algorithm to simulate a point process with unbounded intensity. We extended the idea of Ogata [23] with Kalikow decomposition to introduce numerous new algorithms, see Chapter 4.

1.5.2 State of the works

Chapter 2 corresponds to the paper [27], in collaboration with Alexandre Muzy and Patricia Reynaud-Bouret. This paper is published in Springer Nature Computer Science.

Chapter 3 corresponds to the paper [26]. This paper is submitted to Advances in Applied Probability, which is now in major revision.

The first part of Chapter 4 corresponds to the report of Master internship of Paul Gresland, in which the author is cosupervisor with Alexandre Muzy, Patricia Reynaud-Bouret. The second part of this chapter is a work in progress.

2 Event-Scheduling Algorithms with Kalikow Decomposition for Simulating Potentially Infinite Neuronal Networks

Written by T.C. Phi, A. Muzy and P. Reynaud-Bouret

Status: Published in SN Computer Science.

2.1 Abstract

Event-scheduling algorithms can compute in continuous time the next occurrence of points (as events) of a counting process based on their current conditional intensity. In particular event-scheduling algorithms can be adapted to perform the simulation of finite neuronal networks activity. These algorithms are based on Ogata's thinning strategy [17], which always needs to simulate the whole network to access the behaviour of one particular neuron of the network. On the other hand, for discrete time models, theoretical algorithms based on Kalikow decomposition can pick at random influencing neurons and perform a perfect simulation (meaning without approximations) of the behaviour of one given neuron embedded in an infinite network, at every time step. These algorithms are currently not computationally tractable in continuous time. To solve this problem, an event-scheduling algorithm with Kalikow decomposition is proposed here for the sequential simulation of point processes neuronal models satisfying this decomposition. This new algorithm is applied to infinite neuronal networks whose finite time simulation is a prerequisite to realistic brain modeling.

2.2 Introduction

Point processes in time are stochastic objects that model efficiently event occurrences with a huge variety of applications: time of deaths, earthquake occurrences, gene positions on DNA strand, etc. [1, 20, 22]).

Most of the time, point processes are multivariate [6] in the sense that either several processes are considered at the same time, or in the sense that one process regroups together all the events of the different processes and marks them by their type. A typical example consists in considering either two processes, one counting the wedding events of a given person and one counting the children birth dates of the same person or only one marked process which regroups all the possible dates of birth or weddings independently and adds one mark per point, here wedding or birth.

Consider now a network of neurons each of them emitting action potentials (spikes). These spike trains can be modeled by a multivariate point process with a potentially infinite number of marks, each mark representing one given neuron. The main difference between classical models of multivariate point processes and the ones considered in particular for neuronal networks is the size of the network. A human brain consists in about 10^{11} neurons whereas a cockroach contains already about 10^6 neurons. Therefore the simulation of the whole network is either impossible or a very difficult and computationally intensive task for which particular tricks depending on the shape of the network or the point processes have to be used [5, 13, 19].

Another point of view, which is the one considered here, is to simulate, not the whole network, but the events of one particular node or neuron, embedded in and interacting with the whole network. In this sense, one might consider an infinite network. This is the mathematical point of view considered in a series of papers [8, 9, 18] and based on Kalikow decomposition [11] coupled with perfect simulation theoretical algorithms [4, 7]. However these works are suitable in discrete time and only provide a way to decide at each time step if the neuron is spiking or not. They cannot operate in continuous time, i.e. they cannot directly predict the next event (or spike). Up to our knowledge, there exists only one attempt of using such decomposition in continuous time [10], but the corresponding simulation algorithm is purely theoretical in the sense that the corresponding conditional Kalikow decomposition should exist given the whole infinite realization of a multivariate Poisson process, with an infinite number of marks, quantity which is impossible to simulate in practice.

The aim of the present work is to present an algorithm which

- can operate in continuous time in the sense that it can predict the occurrence of the next event. In this sense, it is an event-scheduling algorithm;
- can simulate the behavior of one particular neuron embedded in a potentially infinite network without having to simulate the whole network;
- is based on an unconditional Kalikow decomposition and in this sense, can only work for point processes with this decomposition.

In Section 2.3, we specify the links between event-scheduling algorithms and the classical point process theory. In Section 2.4, we give the Kalikow decomposition. In Section 2.5, we present the backward-forward perfect simulation algorithm and prove why it almost surely ends under certain conditions. In Section 2.6, we provide simulation results and a conclusion is given in Section 2.7.

2.3 Event-scheduling simulation of point processes

On the one hand, simulation algorithms of multivariate point processes [17] are quite well known in the statistical community but as far as we know quite confidential in the simulation (computer scientist) community. On the other hand, event-scheduling simulation first appeared in the mid-1960s [21] and was formalized as discrete event systems in the mid-1970s [24] to interpret very general simulation algorithms scheduling "next events". A basic event-scheduling algorithm "jumps" from one event occurring at a time stamp $t \in \mathbb{R}_0^+$ to a next event occurring at a next time stamp $t' \in \mathbb{R}_0^+$, with $t' \geq t$. In a discrete event system, the state of the system is considered as changing at times t, t' and conversely unchanging in between [23]. In [13], we have written the main equivalence between the point processes simulation algorithms and the discrete event simulation set-up, which led us to a significant improvement in terms of computational time when huge but finite networks are into play. Usual event-scheduling simulation algorithms have been developed considering independently the components (nodes) of a system. Our approach considers new algorithms for activity tracking simulation [16]. The event activity is tracked from active nodes to children (influencees).

Here we just recall the main ingredients that are useful for the sequel.

To define point processes, we need a filtration or history $(\mathscr{F}_t)_{t\geq 0}$. Most of the time, and this will be the case here, this filtration (or history) is restricted to the internal history of the multivariate process $(\mathscr{F}_t^{int})_{t\geq 0}$, which means that at time $t-$, i.e. just before time t, we only have access to the events that have already occurred in the past strictly before time t, in the whole network. The conditional intensity, $\phi_i(t|\mathscr{F}_{t-}^{int})$, of the point process, representing neuron i gives the instantaneous firing rate, that is the frequency of spikes, given the past contained in $\mathscr{F}_{t-}^{int}$. Let us just mention two very famous examples.

If $\phi_i(t|\mathscr{F}^{int}_{t-})$ is a deterministic constant, say M, then the spikes of neuron i form a homogeneous Poisson process with intensity M. The occurrence of spikes are completely independent from what occurs elsewhere in the network and from the previous occurrences of spikes of neuron i.

If we denote by $\mathbf{I}$ the set of neurons, we can also envision the following form for the conditional intensity:

$$\phi_i(t|\mathscr{F}^{int}_{t-}) = \nu_i + \sum_{j\in\mathbf{I}} w_{ij}(\,\mathbf{Nb}^j_{[t-A,t)} \wedge M). \tag{2.3.1}$$

This is a particular case of generalized Hawkes processes [3]. More precisely ν_i is the spontaneous rate (assumed to be less than the deterministic upper bound $M > 1$) of neuron i. Then every neuron in the network can excite neuron i: more precisely, one counts the number of spikes that have been produced by neuron j just before t, in a window of length A, this is $\mathbf{Nb}^j_{[t-A,t)}$; we clip it by the upper bound M and modulate its contribution to the intensity by the positive synaptic weight between neuron i and neuron j, w_{ij}. For instance, if there is only one spike in the whole network just before time t, and if this happens on neuron j, then the intensity for neuron i becomes $\nu_i + w_{ij}$. The sum over all neurons j mimics the synaptic integration that takes place at neuron i. As a renormalization constraint, we assume that $\sup_{i\in\mathbf{I}}\sum_{j\in\mathbf{I}} w_{ij} < 1$. This ensures in particular that such a process has always a conditional intensity bounded by $2M$.

Hence, starting for instance at time t, and given the whole past, one can compute the next event in the network by computing for each node of the network the next event in absence of other spike apparition. To do so, remark that in absence of other spike apparition, the quantity $\phi_i(s|\mathscr{F}^{int}_{s-})$ for $s > t$ becomes for instance in the previous example

$$\phi^{abs}_i(s,t) = \nu_i + \sum_{j\in\mathbf{I}} w_{ij}(\,\mathbf{Nb}^j_{[s-A,t)} \wedge M),$$

meaning that we do not count the spikes that may occur after t but before s. This can be generalized to more general point processes. The main simulation loop is presented in Algorithm 1

Algorithm 1 Classical point process simulation algorithm

▷ With $[t_0, t_1]$ the interval of simulation
Initialize the family of points $\mathbf{P} = \varnothing$

▷ Each point is a time T with a mark, j_T, which is the neuron on which T appears
Initialize $t \leftarrow t_0$
repeat
    for each neuron $i \in \mathbf{I}$ **do**
        Draw independently an exponential variable E_i with parameter 1
        Apply the inverse transformation, that is, find T_i such that

$$\int_t^{T_i} \phi_i^{abs}(s,t)ds = E_i.$$

    end for
    Compute the time T of the next spike of the system after t, and the neuron where the spike occurs by $T \leftarrow \min_{i \in \mathbf{I}} T_i$, with $j_T \leftarrow \arg\min_{i \in \mathbf{I}} T_i$
    if $T \le t_1$ **then**
        append T with mark j_T to $\mathbf{P}$
    end if
    $t \leftarrow T$
until $t > t_1$

Note that the quantity $\phi_i^{abs}(s,t)$ can be also seen as the hazard rate of the next potential point $T_i^{(1)}$ after t. It is a discrete event approach with the state corresponding to the function $\phi_i^{abs}(.,t)$.

Ogata [17], inspired by Lewis' algorithm [12], added a thinning (also called rejection) step on top of this procedure because the integral $\int_t^{T_i^{(1)}} \phi_i^{abs}(s,t)ds$ can be very difficult to compute. To do so (and simplifying a bit), assume that $\phi_i(t|\mathcal{F}_{t-}^{int})$ is upper bounded by a deterministic constant M. This means that the point process has always less points than a homogeneous Poisson process with intensity M. Therefore Steps 5-6 of Algorithm 1 can be replaced by the generation of an exponential of parameter M, E_i' and deciding whether we accept or reject the point with probability $\phi_i^{abs}(t+E_i',t)/M$. There are a lot of variants of this procedure: Ogata's original one uses actually the fact that the minimum of exponential variables, is still an exponential variable. Therefore one can propose a next point for the whole system, then accept it for the whole system and then decide on which neuron of the network the event is actually appearing. More details on the multivariate Ogata's algorithm can be found in [13].

As we see here, Ogata's algorithm is very general but clearly needs to simulate the whole system to simulate only one neuron. Moreover starting at time t_0, it does not go backward and therefore cannot simulate a Hawkes process in stationary regime. There has been specific algorithms based on clusters representation that aim at perfectly simulate particular univariate Hawkes processes [15]. The algorithm that we propose here, will also overcome this flaw.

2.4 Kalikow decomposition

Kalikow decomposition relies on the concept of neighborhood, denoted by v, which are picked at random and which gives the portion of time and neuron subsets that we need to look at, to move forward. Typically, for a positive constant A, such a v can be:

- $\{(i,[-A,0))\}$, meaning we are interested only by the spikes of neuron i in the window $[-A,0)$;
- $\{(i,[-2A,0)),(j,[-2A,-A))\}$, that is, we need the spikes of neuron i in the window $[-2A,0)$ and the spikes of neuron j in the window $[-2A,-A)$;
- the emptyset $\emptyset$, meaning that we do not need to look at anything to pursue.

We need to also define $l(v)$ the total time length of the neighborhood v whatever the neuron is. For instance, in the first case, we find $l(v)=A$, in the second $l(v)=3A$ and in the third $l(v)=0$.

We are only interested by stationary processes, for which the conditional intensity, $\phi_i(t \mid \mathcal{F}_{t-}^{int})$, only depends on the intrinsic distance between the previous points and the time t and not on the precise value of t *per se*. In this sense the rule to compute the intensity may be only defined at time 0 and then shifted by t to have the conditional intensity at time t. In the same way, the timeline of a neighborhood v is defined as a subset of $\mathbb{R}_-^*$ so that information contained in the neighborhood is included in $\mathcal{F}_{0-}^{int}$, and v can be shifted (meaning its timeline is shifted) at position t if need be. We assume that $\mathbf{I}$ the set of neurons is countable and that we have a countable set of possibilities for the neighborhoods $\mathcal{V}$.

Then, we say that the process admits *a Kalikow decomposition* with bound M and neighborhood family $\mathcal{V}$, if for any neuron $i \in \mathbf{I}$, for all $v \in \mathcal{V}$ there exists a non negative M-bounded quantity ϕ_i^v, which is

$\mathscr{F}_{0-}^{int}$ measurable and whose value only depends on the points appearing in the neighborhood v, and a probability density function $\lambda_i(.)$ such that

$$\phi_i(0 \mid \mathscr{F}_{0-}^{int}) = \lambda_i(\emptyset)\phi_i^{\emptyset} + \sum_{v\in\mathscr{V}, v\neq\emptyset} \lambda_i(v) \times \phi_i^{v} \tag{2.4.1}$$

with $\lambda_i(\emptyset) + \sum_{v\in\mathscr{V}, v\neq\emptyset} \lambda_i(v) = 1$.

Note that because of the stationarity assumptions, the rule to compute the ϕ_i^{v}'s can be shifted at time t, which leads to a predictable function that we call $\phi_i^{v_t}(t)$ which only depends on what is inside v_t, which is the neighborhood v shifted by t. Note also that $\phi_i^{\emptyset}$, because it depends on what happens in an empty neighborhood, is a pure constant.

The interpretation of (2.4.1) is tricky and is not as straightforward as in the discrete case (see [18]). The best way to understand it is to give the theoretical algorithm for simulating the next event on neuron i after time t (cf. Algorithm 2).

Algorithm 2 Kalikow theoretical simulation algorithm

$\triangleright$ With $[t_0, t_1]$ the interval of simulation for neuron i

1: Initialize the family of points $\mathbf{P} = \emptyset$

$\triangleright$ NB: since we are only interested by points on neuron i, $j_T = i$ is a useless mark here.

2: Initialize $t \leftarrow t_0$
3: **repeat**
4:     Draw an exponential variable E with parameter M, and compute $T = t + E$.
5:     Pick a random neighborhood according to the distribution $\lambda_i(.)$ given in the Kalikow decomposition and shift the neighborhood at time T: this is V_T.
6:     Draw X_T a Bernoulli variable with parameter $\frac{\phi_i^{V_T}(T)}{M}$
7:     **if** $X_T = 1$ and $T \leq t_1$ **then**
8:         append T to $\mathbf{P}$
9:     **end if**
10:     $t \leftarrow T$
11: **until** $t > t_1$

This Algorithm is close to Algorithm 1 but adds a neighborhood choice (Step 5) with a thinning step (Steps 6-9).

In Appendix 2.8, we prove that this algorithm indeed provides a point process with an intensity given by (2.4.1) shifted at time t.

The previous algorithm cannot be put into practice because the computation of $\phi_i^{V_T}$ depends on the points in V_T, that are not known at this stage. That is why the efficient algorithm that we propose in the next section goes backward in time before moving forward.

2.5 Backward Forward algorithm

Let us now describe the complete *Backward Forward algorithm* (cf. Algorithm 3). Note that to do so, the set of points $\mathbf{P}$ is not reduced, as in the two previous algorithms, to the set of points that we want to simulate but this contains all the points that need to be simulated to perform the task.

Algorithm 3 Backward Forward Algorithm

▷ With $[t_0, t_1]$ the interval of simulation for neuron $i \in \mathbf{I}$

Init.

Initialize the family $\mathbf{V}$ of non empty neighborhoods with $\{(i, [t_0, t_1])\}$

Initialize the family of points $\mathbf{P} = \emptyset$

▷ Each point is a time T with 3 marks: j_T is the neuron on which T appears, V_T for the choice of neighborhood, X_T for the thinning step (accepted/rejected)

Draw E an exponential variable with parameter M

Schedule $T_{next} = t_0 + E$

while $T_{next} < t_1$ **do**

Init. Marks

    Append to $\mathbf{P}$, the point T_{next}, with 3 marks: $j_{T_{next}} = i$, $V_{T_{next}} =$ n.a. and $X_{T_{next}} =$ n.a. (n.a. stand for *not assigned yet*)

Back-ward

    while There are points T in $\mathbf{P}$ with $V_T =$ n.a. **do**

        for each point T in $\mathbf{P}$ with $V_T =$ n.a. **do**

            Update V_T by drawing V_T according to λ_{j_T} shifted at time T.

            if $V_T \neq \emptyset$ **then**

                Find the portion of time/neurons in V_T which does not intersect the existing non empty neighborhoods in $\mathbf{V}$

                Simulate on it a Poisson process with rate M

                Append the simulated points, T', if any, to $\mathbf{P}$ with their neuron $j_{T'}$ and with $V_{T'} = X_{T'} =$n.a

                Append V_T to $\mathbf{V}$

            end if

        end for

    end while

Forward

    Sort the T's in $\mathbf{P}$ with $X_T =$n.a. in increasing order

    for each of them starting with the most backward **do**

        Draw X_T as a Bernoulli variable with parameter $\frac{\phi_{j_T}^{V_T}(T)}{M}$

    end for

    Draw E' another exponential variable with parameter M

    $T_{next} \leftarrow T_{next} + E'$

end while

The desired points are the points in $\mathbf{P}$ with marks $j_T = i$, $X_T = 1$ and that appear in $[t_0, t_1]$

▷ It is possible that the algorithm generated points before t_0 and they have to be removed

---

At the difference with Algorithm 2, the main point is that in the backward part we pick at random all the points that may influence the thinning step. The fact that this loop ends comes from the following Proposition.

Proposition 1. If

$$\sup_{i\in \mathbf{I}} \sum_{v\in\mathcal{V}} \lambda_i(v) l(v) M < 1. \tag{2.5.1}$$

then the backward part of Algorithm 3 ends almost surely in finite time.

The proof is postponed to Appendix 2.9. It is based on branching process arguments. Basically if in Steps 8-16, we produce in average less than one point, either because we picked the empty set in V_T or because the simulation of the Poisson process ended up with a small amount of points, eventually none, then the loop ends almost surely because there is an extinction of the corresponding branching process.

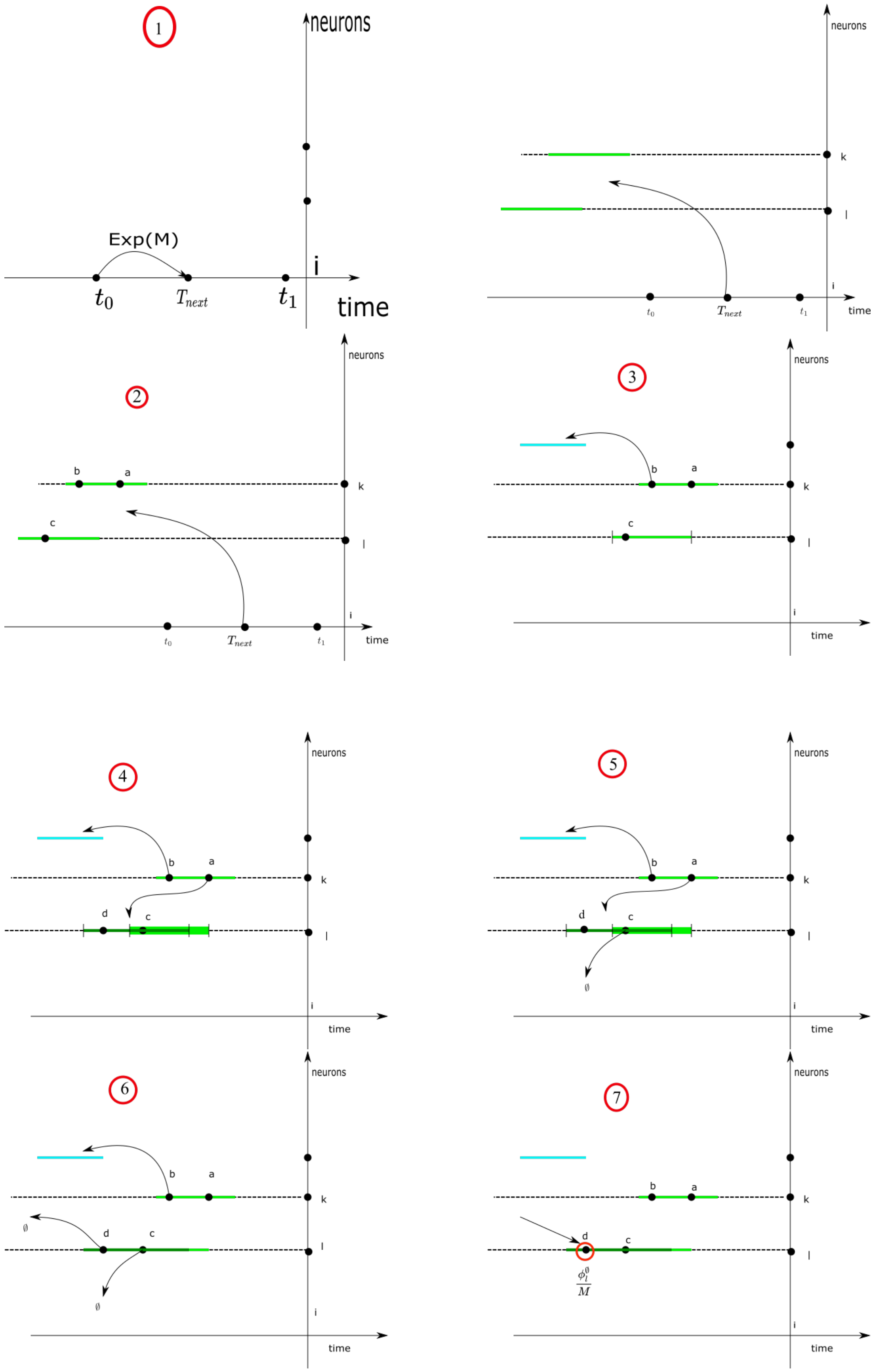
In the backward part, one of the most delicate part consists in being sure that we add new points only if we have not visited this portion of time/neurons before (see Steps 11-13). If we do not make this verification, we may not have the same past depending on the neuron we are looking at and the procedure would not simulate the process we want.

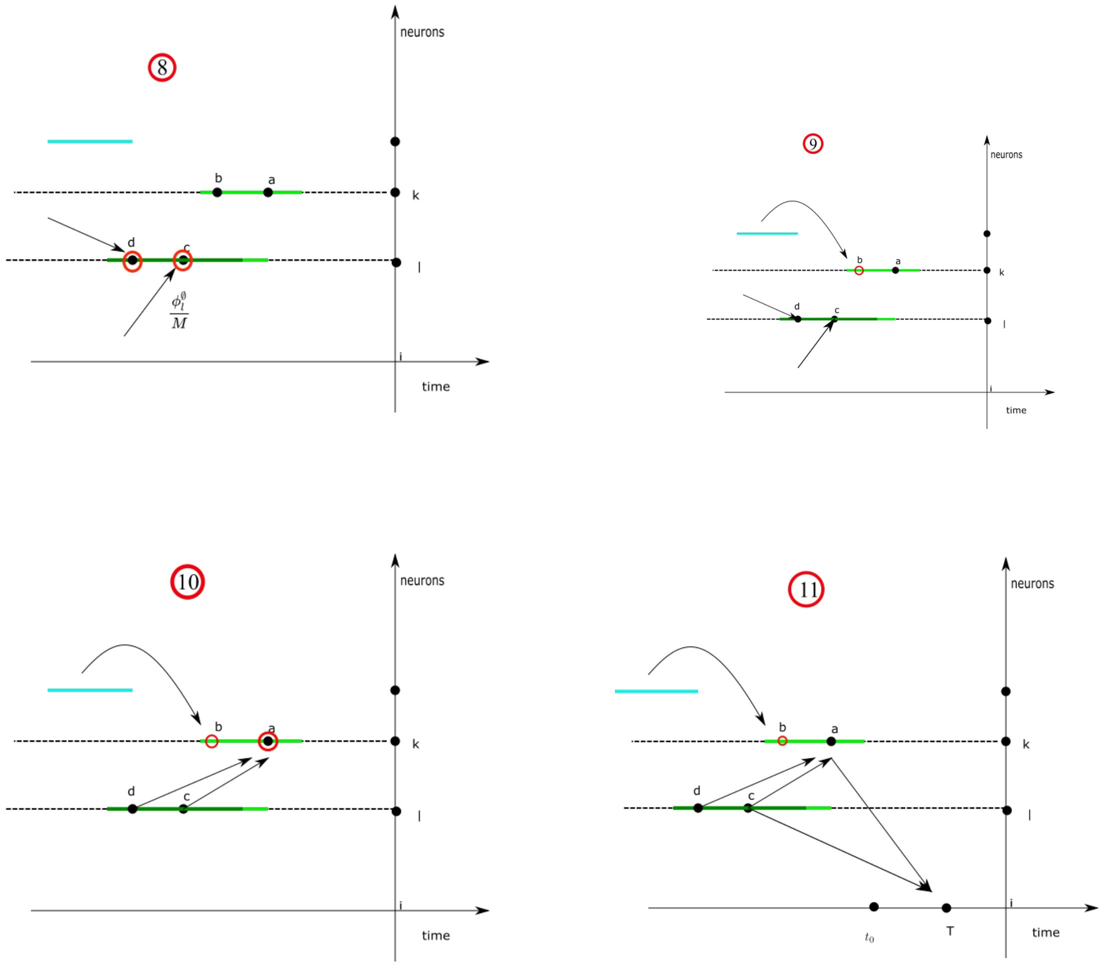
In the forward part, because the backward algorithm stopped just before, we are first sure to have assess all V_T's. Since $\phi_j^{V_t}(t)$ is $\mathcal{F}_{t-}^{int}$ measurable, for all t, $\phi_{j_T}^{V_T}(T)$ only depends on the points in $\mathbf{P}$ with mark $X_T = 1$ inside V_T. The problem in Algorithm 2, phrased differently, is that we do not know the marks X_T of the previous points when we have to compute $\phi_{j_T}^{V_T}(T)$. But in the forward part of Algorithm 3, we are sure that the most backward point for which the thinning (X_T =n.a.) has not taken place, satisfies

- either $V_T = \emptyset$
- or $V_T \neq \emptyset$ but either there are no simulated points in the corresponding V_T or the points there come from previous rounds of the loop (Step 5). Therefore their marks X_T have been assigned.

Therefore, with the Backward Forward algorithm, and at the difference to Algorithm 2, we take the points in an order for which we are sure that we know the previous needed marks.

Figure 4 describes an example to go step by step through Algorithm 3. The *backward steps* determine all the points that may influence the acceptation/rejection of point T_{next}. Notice that whereas usual activity tracking algorithms for point processes [13] automatically detect the active children (influencees), activity tracking in the backward steps detect the parents (influencers). The *forward steps* finally select the points.


1
neurons
Exp(M)
i
t_0
T_{next}
t_1
time
neurons
k
l
i
t_0
T_{next}
t_1
time
neurons
2
b
a
k
c
l
i
t_0
T_{next}
t_1
time
neurons
3
b
a
k
c
l
i
time
neurons
4
b
a
k
d
c
l
i
time
neurons
5
b
a
k
d
c
l
∅
i
time
neurons
6
b
a
k
∅
d
c
l
∅
i
time
neurons
7
b
a
k
d
c
l
$\frac{\phi_l^\emptyset}{M}$
i
time



Remark 4. Main flow example for Algorithm 3, with backward steps (cf. Algorithm 3, Steps 7-17) and forward steps (cf. Algorithm 3, Steps 18-25). Following circled numbers: (1) The next point $T_{next} = t_0 + E$ (cf. Algorithm 3, Step 4) is scheduled, (2) The neighborhood $V_{T_{next}}$ is selected in the first backward step, a *first generation* of three points (a, b on neuron k and c on neuron ℓ) is drawn (cf. Algorithm 3, Step 9), thanks to a Poisson process, (cf. Algorithm 3, Steps 11-12) and appended to **P** (cf. Algorithm 3, Step 13), (3) at the *second generation*, a non empty neighborhood is found, *i.e.* $V_b \neq \emptyset$ (cf. Algorithm 3, Steps 9-1), but the Poisson process simulation does not give any point in it (cf. Algorithm 3, Step 12), (4) at the *second generation*, the neighborhood V_a is picked, it is not empty and overlap the neighborhood of the first generation (cf. Algorithm 3, Steps 9-11): therefore there is no new simulation in the overlap (c is kept and belongs to V_b as well as V_a) but there is a new simulation thanks to a Poisson process outside of the overlap leading to a new point d (cf. Algorithm 3, Step 12)(5) at the *second generation*, for point c, one pick the empty neighborhood, *i.e.* $V_c = \emptyset$ (cf. Algorithm 3, Step 9) and therefore we do not simulate any Poisson process, (6) at *third generation*, similarly no point and no interval are generated, *i.e.* $V_d = \emptyset$ (cf. Algorithm 3, Step 9). This is the end of the backward steps and the beginning of the forward ones, (7)

the point d is not selected, acceptation/selection taking place with probability $\frac{\phi_\ell^\varnothing}{M}$ (cf. Algorithm 3, Step 20), (8) the point c is accepted, here again with probability $\frac{\phi_\ell^\varnothing}{M}$ (cf. Algorithm 3, Step 20), (9) the point b is not selected, acceptation taking place, here, with probability $\frac{\phi_k^{V_b}(b)}{M}$ (cf. Algorithm 3, Step 20), (10) the point a is selected, acceptation taking place, here, with probability $\frac{\phi_k^{V_a}(a)}{M}$ (cf. Algorithm 3, Step 20), (11) The neighborhood of neuron i contains two points, one on neuron k and one on neuron ℓ and one selects T_{next} with probability $\frac{\phi_i^{V_{T_{next}}}(T_{next})}{M}$.

2.6 Illustration

To illustrate in practice the algorithm, we have simulated a Hawkes process as given in (2.3.1). Indeed such a process has a Kalikow decomposition (2.4.1) with bound M and neighborhood family $\mathcal{V}$ constituted of the v's of the form $v = \{(j, [-A, 0))\}$ for some neuron j in $\mathbf{I}$. To do that, we need the following choices:

$$\lambda_i(\emptyset) = 1 - \sum_{j \in \mathbf{I}} w_{ij} \quad \text{and} \quad \phi_i^{\emptyset} = \frac{\nu_i}{\lambda_i(\emptyset)}$$

and for v of the form $v = \{(j, [-A, 0))\}$ for some neuron j in $\mathbf{I}$,

$$\lambda_i(v) = w_{ij} \quad \text{and} \quad \phi_i^v = \mathbf{Nb}^j_{[-A,0)} \wedge M.$$

We have taken $\mathbf{I} = \mathbb{Z}^2$ and the w_{ij} proportional to a discretized centred symmetric bivariate Gaussian distribution of standard deviation σ. More precisely, once $\lambda_i(\emptyset) = \lambda_{\emptyset}$ fixed, picking according to λ_i consists in

- choosing whether V is empty or not with probability $\lambda_{\emptyset}$
- if $V \neq \emptyset$, choosing $V = \{(j, [-A, 0))\}$ with $j - i = \text{round}(W)$ and W obeys a bivariate $\mathcal{N}(0, \sigma^2)$. The round function returns the integer number that is closest to W.

In all the sequel, the simulation is made for neuron $i = (0, 0)$ with $t_0 = 0$, $t_1 = 100$ (see Algorithm 3). The parameters M, $\lambda_{\emptyset}$ and σ vary. The parameters $\nu_i = \nu$ and A are fixed accordingly by

$$\nu = 0.9 M \lambda_{\emptyset} \text{ and } A = 0.9 M^{-1} (1 - \lambda_{\emptyset})^{-1},$$

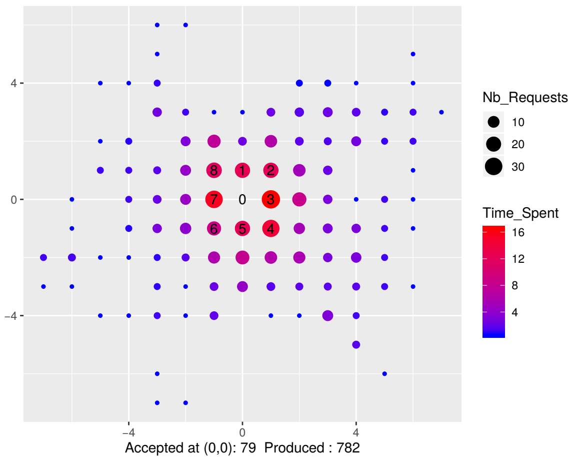
to ensure that $\phi_i^{\emptyset} < M$ and (2.5.1), which amounts here to $(1 - \lambda_{\emptyset}) A M < 1$.

On Figure 2.4(a), with $M = 2$, $\sigma = 1$ and $\lambda_{\emptyset}$ small, we see the overall spread of the algorithm around the neuron to simulate (here $(0, 0)$). Because we chose a Gaussian variable with small variance for the λ_i's, the spread is weak and the neurons very close to the neuron to simulate are requested a lot of time at Steps 9-11 of the algorithm. This is also where the algorithm spent the biggest amount of time to simulate Poisson processes. Note also that roughly to simulate 80 points, we need to simulate 10 times more points globally in the infinite network. Remark also on Figure 2.4(b), the avalanche phenomenon, typical of Hawkes processes: for instance the small cluster of black points on neuron with label 0 (i.e. (0,0)) around time 22, is likely to be due to an excitation coming for the spikes generated (and accepted) on neuron labeled 8 and self excitation. The beauty of the algorithm is that we do not need to have the whole time line of neuron 8 to trigger neuron 0, but only the small blue pieces: we just request them at random, depending on the Kalikow decomposition.

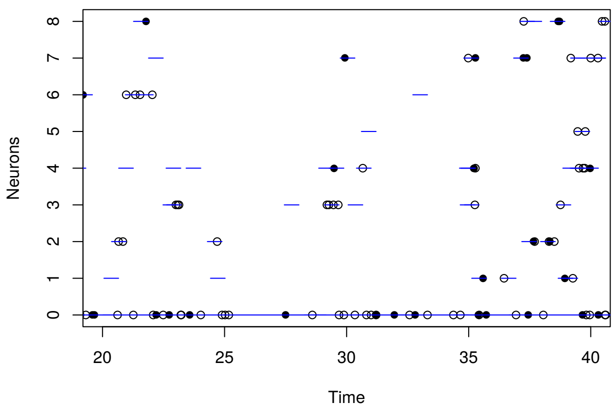
On Figure 2.5, we can first observe that when the parameter which governs the range of λ_i's increase, the global spread of the algorithm increase. In particular, comparing the top left of Figure 2.5 to Figure 2.4 where the only parameter that changes is σ, we see that the algorithm is going much further away and simulates much more points for a sensible equal number of points to generate (and accept) on neuron (0,0). Moreover we can observe that

- From left to right, by increasing $\lambda_{\emptyset}$, it is more likely to pick an empty neighborhood and as a consequence, the spread of the algorithm is smaller. By increasing $\nu = 0.9M\lambda_{\emptyset}$, this also increases the total number of points produced on neuron (0;0).

- From top to bottom, by increasing M, there are more points which are simulated in the Poisson processes (Step 12 of Algorithm 3) and there is also a stronger interaction (we do not truncate that much the number of points in ϕ^{υ}). Therefore, the spread becomes larger and more uniform too, because there are globally more points that are making requests. Moreover, by having a basic rate M which is 10 times bigger, we have to simulate roughly 10 times more points.



(a) Summary of one simulation



(b) Extract of the time simulation for neurons 0 to 9

Figure 2.4: Simulation for $M = 2$, $\sigma = 1$, $\lambda_{\emptyset} = 0.25$. For each neuron in $\mathbb{Z}^2$, that have been requested in Steps 9:11, except the neuron of interest $(0,0)$, have been counted the total number of requests, that is the number of time a V_T pointed towards this neuron (Steps 9 and 11) and the total time spent at this precise neuron simulating a homogeneous Poisson process (Step 12). Note that since the simulation is on $[0,100]$ the time spent at position $(0,0)$ is at least 100. On (a), the summary for one simulation with below the plot, number of points accepted at neuron $(0,0)$ and total number of points that have been simulated. Also annotated on (a), with labels between 0 and 8, the 9 neurons for which the same simulation in [20,40] is represented in more details on (b). More precisely on (b), in abscissa is time and the neuron labels in ordinate. A plain dot represents an accepted point, by the thinning step (Step 20 of Algorithm 3), and an empty round, a rejected point. The blue pieces of line represent the non empty neighborhoods that are in $\mathbf{V}$.

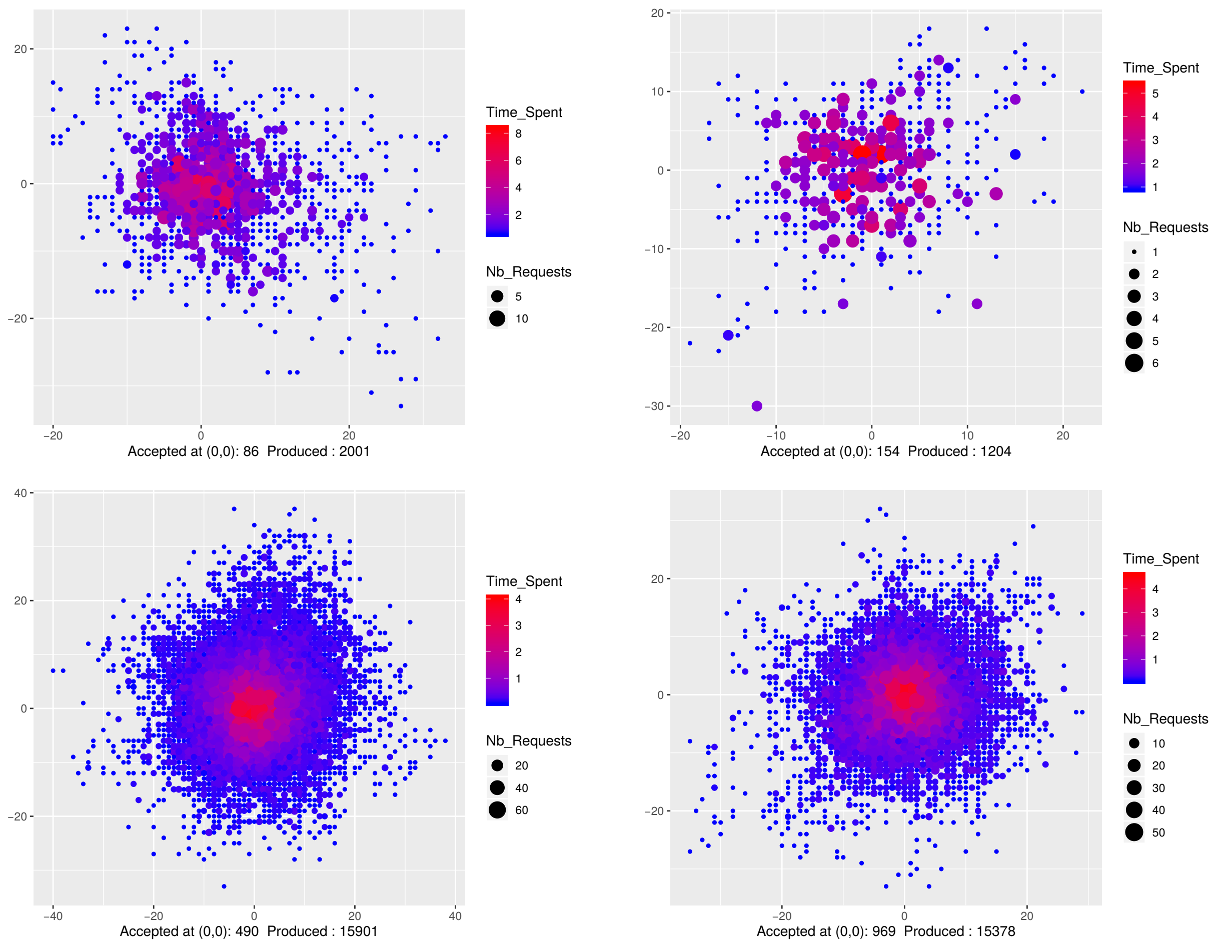


Figure 2.5: Simulation for 4 other sets of parameters, all of them with $\sigma = 3$. Summaries as explained in Figure 2.4. On top, $M = 2$; on bottom, $M = 20$. On the left part, $\lambda_{\emptyset} = 0.25$, on the right part, $\lambda_{\emptyset} = 0.5$.

2.7 Conclusion

We derived a new algorithm for simulating the behavior of one neuron embedded in an infinite network. This is possible thanks to the Kalikow decomposition which allows picking at random the influencing neurons. As seen in the last section, it is computationnally tractable in practice to simulate open systems in the physical sense. A question that remains open for future work is whether we can prove that such a decomposition exists for a wide variety of processes, as it has been shown in discrete time (see [8, 9, 18]).

2.8 Link between Algorithm 2 and the Kalikow decomposition

To prove that Algorithm 2 returns the desired processes, let us use some additional and more mathematical notation. Note that all the points simulated on neuron i before being accepted or not can be seen as coming from a common Poisson process of intensity M, denoted Π_i. For any $i \in \mathbf{I}$, we denote the arrival times of Π_i, $(T_n^i)_{n\in\mathbb{Z}}$, with T_1^i being the first positive time.

As in Step 6 of Algorithm 2, we attach to each point of Π^i a stochastic mark X given by,

$$X_n^i = \begin{cases} 1 & \text{if} \quad T_n^i \text{ is accepted in the thinning procedure} \\ 0 & otherwise. \end{cases} \tag{2.8.1}$$

Let us also define V_n^i the neighborhood choice of T_n^i picked at random and independently of anything else according to λ_i and shifted at time T_n^i.

In addition, for any $i \in \mathbf{I}$, define $N^i = (T_n^i, X_n^i)_{n\in\mathbb{Z}}$ an E-marked point process with $E = \{0; 1\}$. In particular, following the notation in Chapter VIII of [2], for any $i \in \mathbf{I}$, let

$$N_t^i(mark) = \sum_{n\in\mathbb{Z}} \mathbb{1}_{X_n^i=mark}\mathbb{1}_{T_n^i\leq t} \quad \text{for} \quad mark \in E$$

$$\mathscr{F}_t^N = \bigvee_{i\in\mathbf{I}} \sigma(N_s^i(0), N_s^i(1); s \leq t) \quad \text{and} \quad \mathscr{F}_t^{N(1)} = \bigvee_{i\in\mathbf{I}} \sigma(N_s^i(1); s \leq t).$$

Moreover note that $(N_t^i(1))_{t\in\mathbb{R}}$ is the counting process associated to the point processs $\mathbf{P}$ simulated by Algorithm 2. Let us denote by $\varphi_i(t)$, the formula given by (2.4.1) and shifted at time t. Note that since the ϕ_i^v's are $\mathscr{F}_{0-}^{int} = \mathscr{F}_{0-}^{N(1)}$, $\varphi_i(t)$ is $\mathscr{F}_{t-}^{N(1)}$ measurable. We also denote $\varphi_i^v(t)$ the formula of ϕ_i^v shifted at time t.

With this notation, we can prove the following.

Proposition 2. The process $(N_t^i(1))_{t\in\mathbb{R}}$ admits $\varphi_i(t)$ as $\mathscr{F}_t^{N(1)}$-predictable intensity.

Proof. Following the technique in Chapter 2 of [2], let us take C_t a non negative predictable function with respect to (w.r.t) $\mathscr{F}_t^{N^i(1)}$ thatis $\mathscr{F}_{t-}^{N(1)}$ measurable and therefore $\mathscr{F}_{t-}^{N}$ measurable . We have, for any $i \in \mathbf{I}$,

$$\mathbb{E}\left(\int_0^\infty C_t d N_t^i(1)\right) = \sum_{n=1}^\infty \mathbb{E}\Big(C_{T_n^i}\mathbb{1}_{X_n^i=1}\Big)$$

Note that by Theorem T35 at Appendix A1 of [2], any point T should be understood as a stopping time, and that by Theorem T30 at Appendix A2 of [2],

$$\mathscr{F}_{T-}^N = \bigvee_j \sigma\{T_m^j, X_m^j \text{ such that } T_m^j < T\}$$

So

$$\mathbb{E}\left(\int_0^\infty C_t d N_t^i(1)\right) = \sum_{n=1}^\infty \mathbb{E}\Big(C_{T_n^i}\mathbb{E}(\mathbb{1}_{X_n^i=1}|\mathscr{F}_{T_n^i-}^N, V_n^i)\Big) = \sum_{n=1}^\infty \mathbb{E}\left(C_{T_n^i}\frac{\varphi_i^{V_n^i}(T_n^i)}{M}\right).$$

Let us now integrate with respect to the choice V_n^i, which is independent of anything else.

$$\mathbb{E}\left(\int_0^\infty C_t d N_t^i(1)\right) = \sum_{n=1}^\infty \mathbb{E}\left(C_{T_n^i}\frac{\lambda_i(\emptyset)\varphi_i^\emptyset + \sum_{v\in\mathscr{V}, v\neq\emptyset}\lambda_i(v)\times\varphi_i^v(T_n^i)}{M}\right) = \mathbb{E}\left(\int_0^\infty C_t \frac{\varphi_i(t)}{M} d\Pi^i(t)\right).$$

Since Π^i is a Poisson process with respect to $(\mathcal{F}_t^N)_t$ with intensity M, and since $C_t \frac{\varphi_i(t)}{M}$ is $\mathcal{F}_{t-}^N$ measurable, we finally have that

$$\mathbb{E}\left(\int_0^\infty C_t dN_t^i(1)\right) = \mathbb{E}\left(\int_0^\infty C_t \varphi_i(t) dt\right),$$

which ends the proof. □

2.9 Proof of Proposition 1

Proof. We do the proof for the *backward* part, starting with $T = T_{next}$ as the next point after t_0 (Step 4 of Algorithm 3), the proof being similar for the other T_{next} generated at Step 23. We construct a tree with root (i, T). For each point $(j_{T'}, T')$ in the tree, the points which are simulated in $V_{T'}$ (Step 12 of Algorithm 3) define the children of $(j_{T'}, T')$ in the tree. This forms the tree $\tilde{\mathcal{T}}$.

Let us now build a tree $\tilde{\mathcal{C}}$ with root (i, T) (that includes the previous tree) by mimicking the previous procedure in the *backward* part, except that we simulate on the whole neighborhood even if it has a part that intersects with previous neighborhoods (if they exist) (Step 11-12 of Algorithm 3). By doing so, we make the number of children at each node independent of anything else.

If the tree $\tilde{\mathcal{C}}$ goes extinct then so does the tree $\tilde{\mathcal{T}}$ and the backward part of the algorithm terminates.

But if one only counts the number of children in the tree $\tilde{\mathcal{C}}$, we have a marked branching process whose reproduction distribution for the mark i is given by

- no children with probability $\lambda_i(\emptyset)$
- Poissonian number of children with parameter $l(v)M$ if v is the chosen neighborhood with probability $\lambda_i(v)$

This gives that the average number of children issued from a node with the mark i is

$$\zeta_i = \lambda_i(\emptyset) \times 0 + \sum_{v \in \mathcal{V}, v \neq \emptyset} \lambda_i(v) l(v) M.$$

If we denote $\tilde{\mathcal{C}}^k$ as the collection of points in the tree $\tilde{\mathcal{C}}$ at generation k, and by $K_{T'}$ the set of points generated independently as a Poisson process of rate M inside $V_{T'}$, we see recursively that

$$\tilde{\mathcal{C}}^{k+1} = \bigcup_{T' \in \tilde{\mathcal{C}}^k} K_{T'}$$

But

$$\mathbb{E}(|K_{T'}| | T') = \zeta_{j_{T'}}.$$

Therefore, if we denote the total number of sites in $\tilde{\mathcal{C}}^k$ by $Z^{(k)}$, we have

$$\mathbb{E}(Z^{(k+1)} | \tilde{\mathcal{C}}^k) \leq Z^{(k)} \sup_{i \in I} \zeta_i.$$

One can then conclude by recursion that,

$$\mathbb{E}(Z^{(k)}) \leq (\sup_{i \in I} \zeta_i)^k < 1.$$

The last inequality use the sparsity neighborhood assumption. Then we deduce that, the mean number of children in each generation goes to 0 as k tends to infinity. So by using classical branching techniques in [14], we conclude that the tree $\tilde{\mathscr{C}}$ will go extinct almost surely. This also implies that, the backward steps end a.s.

□

Acknowledgements

This work was supported by the French government, through the UCA^{Jedi} Investissements d'Avenir managed by the National Research Agency (ANR-15-IDEX-01) and by the interdisciplinary Institute for Modeling in Neuroscience and Cognition (NeuroMod) of the Université Côte d'Azur. The authors would like to thank Professor E.Löcherbach from Paris 1 for great discussions about Kalikow decomposition and Forward Backward Algorithm.

References for Chapter 2

1. P. K. Andersen, O. Borgan, R. D. Gill, and N. Keiding. *Statistical models based on counting processes*. Springer Science & Business Media, 2012.
2. P. Brémaud. *Point processes and queues: martingale dynamics*. Vol. 50. Springer, 1981.
3. P. Brémaud and L. Massoulié. "Stability of nonlinear Hawkes processes". *The Annals of Probability*, 1996, pp. 1563–1588.
4. F. Comets, R. Fernández, and P. A. Ferrari. "Processes with long memory: regenerative construction and perfect simulation". *The Annals of Applied Probability* 12:3, 2002, pp. 921–943.
5. A. Dassios and H. Zhao. "Exact simulation of Hawkes process with exponentially decaying intensity". *Electronic Communications in Probability* 18, 2013, pp. 1–13.
6. V. Didelez. "Graphical models for marked point processes based on local independence". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70:1, 2008, pp. 245–264.
7. R. Fernández, P. Ferrari, and A. Galves. "Coupling, renewal and perfect simulation of chains of infinite order". *Lecture Notes for the vth Brazilian school of Probability, Ubatuba* 2001, 2001.
8. A. Galves and E. Löcherbach. "Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets". *Journal of Statistical Physics* 151:5, 2013, pp. 896–921.
9. A. Galves and E. Löcherbach. "Modeling networks of spiking neurons as interacting processes with memory of variable length". *Journal de la Société Française de Statistique* 157:1, 2016, pp. 17–32.
10. P. Hodara and E. Löcherbach. "Hawkes processes with variable length memory and an infinite number of components". *Advances in Applied Probability* 49:1, 2017, pp. 84–107.

11. S. Kalikow. “Random Markov processes and uniform martingales”. *Israel Journal of Mathematics* 71:1, 1990, pp. 33–54.
12. P. W. Lewis and G. S. Shedler. “Simulation of nonhomogeneous Poisson processes by thinning”. *Naval research logistics quarterly* 26:3, 1979, pp. 403–413.
13. C. Mascart, A. Muzy, and P. Reynaud-Bouret. “Efficient Simulation of Sparse Graphs of Point Processes”. *arXiv preprint arXiv:2001.01702*, 2020.
14. S. Méléard. *Aléatoire: Introduction à la théorie et au calcul des probabilités*. Editions Ecole Polytechnique, 2010.
15. J. Møller and J. G. Rasmussen. “Perfect simulation of Hawkes processes”. *Advances in applied probability* 37:3, 2005, pp. 629–646.
16. A. Muzy. “Exploiting activity for the modeling and simulation of dynamics and learning processes in hierarchical (neurocognitive) systems”. *Computing in Science & Engineering* 21:1, 2019, pp. 84–93.
17. Y. Ogata. “On Lewis’ simulation method for point processes”. *IEEE transactions on information theory* 27:1, 1981, pp. 23–31.
18. G. Ost and P. Reynaud-Bouret. “Sparse space–time models: Concentration inequalities and Lasso”. In: *Annales de l’Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 56. 4. Institut Henri Poincaré. 2020, pp. 2377–2405.
19. E. A. Peters et al. “Rejection-free Monte Carlo sampling for general potentials”. *Physical Review E* 85:2, 2012, p. 026703.
20. P. Reynaud-Bouret and S. Schbath. “Adaptive estimation for Hawkes processes; application to genome analysis”. *The Annals of Statistics* 38:5, 2010, pp. 2781–2822.
21. K. Tocher. “PLUS/GPS III Specification”. *United Steel Companies Ltd, Department of Operational Research*, 1967.
22. D. Vere-Jones and T. Ozaki. “Some examples of statistical estimation applied to earthquake data”. *Annals of the Institute of Statistical Mathematics* 34:1, 1982, pp. 189–207.
23. B. P. Zeigler, A. Muzy, and E. Kofman. *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018.
24. B. Zeigler. “Theory of modeling and simulation. Jhon Wiley & Sons”. *Inc., New York, NY*, 1976.

3 Kalikow decomposition for counting processes with stochastic intensity

Written by T.C. Phi

Status: Submitted and in revision in Advances in Applied Probability

3.1 Abstract

We propose a new Kalikow decomposition and the corresponding Perfect Simulation algorithm for continuous time multivariate counting processes, on potentially infinite networks. We prove the existence of such a decomposition in various cases. This decomposition is not unique and we discuss the choice of the decomposition in terms of algorithmic efficiency. We apply these methods on several examples: linear Hawkes process, age dependent Hawkes process, exponential Hawkes process.

3.2 Introduction

Multivariate point (or counting) processes on networks have been used to model a large variety of situations : social networks [11], financial prices [2], genomics [22], etc. One of the most complex network models comes from neuroscience where the number of nodes can be as large as billions [16, 19, 23]. Several counting process models have been used to model such large networks: Hawkes processes [12, 13], Galves-Löcherbach models [10] etc. If the simulation of such large and potentially infinite networks is of fundamental importance in computational neuroscience [16, 19], also the existence of such processes in stationary regime and within a potentially infinite network draws a lot of interest (see [10, 14, 18] in discrete or continuous time).

Kalikow decompositions [15] have been introduced and mainly used in discrete time. Such a decomposition provides a decomposition of the transition probabilities into a mixture of more elementary transitions. The whole idea is that even if the process is complex (infinite memory, infinite network), the elementary transitions look only at what happens in a finite neighborhood in time and space. Once the decomposition is proved for a given process, this can be used to write a Perfect Simulation algorithm [10, 14, 18]. Here the word "perfect" refers to the fact that it is possible in finite time to simulate what happens on one of the nodes of the potentially infinite network in a stationary regime. The existence of such an algorithm guarantees in particular the existence of the process in stationary regime. Most of the papers referring to Kalikow decomposition and Perfect Simulation are theoretical and aim at proving the existence of such processes on infinite networks in stationary regime [10, 14].

In the present paper, we propose to go from discrete to continuous time. Therefore, we will decompose conditional intensities rather than transition probabilities. This leads to serious difficulties that usually prevent a more practical application of the Perfect Simulation algorithm. Indeed, up to our knowledge, the only work dealing with continuous time counting processes, is the one by Hodara and Löcherbach [14]. Their decomposition is constructed under the assumption that there is a dominating Poisson process on each of the nodes, from which the points of the processes under interest can be thinned by rejection sampling (see also [17] for another use of thinning in simulation of counting processes). To prove the existence of a Kalikow decomposition and go back to a more classic discrete time setting, the authors need to freeze the dominating Poisson process, leading to a mixture, in the Kalikow decomposition, that depends on the realization of the dominating Poisson process. Such a mixture is not accessible in practice, and this prevents the use of their Perfect Simulation algorithm for more concrete purposes than mere existence.

More recently, in a previous computational article [19], we have used another type of Kalikow decomposition, which does not depend on the dominating Poisson process. This leads to a Perfect Simulation algorithm, which can be used as a concrete way for Computational Neuroscience to simulate neuronal networks as an open physical system, where we do not need to simulate the whole network to simulate what happens in a small part [19].

In the present work, we want to go further, by proposing an even more general Kalikow decomposition, which does not assume the existence of a dominating Poisson process at all. We also prove (and this is not done in [19]) that such decomposition exists on various interesting examples, even if these decompositions are not unique. Finally we propose a corresponding Perfect Simulation algorithm and we discuss its efficiency with respect to the decomposition that is used.

The paper is organized as follows. In Section 3.3, we introduce the basic notation and give the precise definition of a Kalikow decomposition. In Section 3.4, we present two methods to obtain a Kalikow decomposition for a counting process having stochastic intensity. The first method is very general and based on adequate choices of neighborhoods and weights. As an application, we study a classical example, the linear Hawkes process [12, 13], and a very recent and promising process, the age dependent Hawkes process [20]. Though very simple cases of Hawkes processes are treated, it is worth to note that our method can be generalized very easily. The second method exists only for Hawkes processes and is based on Taylor expansion. Finally, in Section 3.5, we present a modified Perfect Simulation algorithm based on the Kalikow decomposition written in Section 3.4, and we discuss the efficiency of the algorithm with respect to the Kalikow decomposition.

3.3 Notation and Kalikow decomposition

3.3.1 Notation and Definition

We start this section by recalling the definition of counting processes and stochastic intensity. We refer the reader to [3] and [7] for more complete statements.

Let $\mathbf{I}$ be a countable index set. A counting process $Z^i, i \in \mathbf{I}$, can be described by its sequence of jump times in $\mathbb{R}$, $(T^i_n)_{n\in\mathbb{Z}}$ [3]. Consider $(\mathcal{F}_t)_{t\in\mathbb{R}}$ the past filtration of the process $Z = (Z^i)_{i\in\mathbf{I}}$:

$$\mathcal{F}_t = \sigma(Z^i_s, i \in \mathbf{I}, s \leq t).$$

Since a point process is fully characterized by its arrival times [3], we can denote by $\mathcal{X}$ the canonical path space of Z:

$$\mathcal{X} = \{(\{t^i_n\}_{n\in\mathbb{Z}})_{i\in\mathbf{I}} \quad \text{such that} \,\forall n, i, \quad t^i_n < t^i_{n+1} \text{ and } t^i_0 \leq 0 < t^i_1\},$$

where $\{t^i_n\}_{n\in\mathbb{Z}}$ denotes a possible realization of $(T^i_n)_{n\in\mathbb{Z}}$. Denote $\mathcal{X}_t$ the canonical path space of Z before time t:

$$\mathcal{X}_t = \mathcal{X} \cap (-\infty, t)^I.$$

A past configuration x_t is an element of $\mathcal{X}_t$ which is a realization of arrival times of Z before t.

Under suitable assumptions, the evolution of the point process Z^i with respect to $(\mathcal{F}_t)_{t\in\mathbb{R}}$ is fully characterized by its stochastic intensity which depends on the past configuration, see Proposition 7.2.IV of [7]. Hence, in this paper, for any $x_t \in \mathcal{X}_t$, given that the past before time t is x_t, we denote by $\phi_{i,t}(x_t)$ the corresponding stochastic intensity of the process Z^i at time t for any $i \in \mathbf{I}$. More precisely, for any $x_t \in \mathcal{X}_t$, we have

$$\mathbb{P}\big(Z^i \text{ has jump in } [t, t+dt) \mid \text{past before time } t = x_t\big) = \phi_{i,t}(x_t)dt.$$

Together with the path space $\mathcal{X}_t$, we denote $\mathbf{V}_t$ a countable collection of finite space-time neighborhoods v_t, in which each neighborhood v_t is a Borel subset of $\mathbf{I} \times (-\infty, t)$. More precisely, we call v_t a finite neighborhood if there exists a finite subset $J \subset \mathbf{I}$ and a finite interval $[a, b]$ such that:

$$v_t \subset J \times [a, b].$$

For convenience, we denote $\mathcal{X}$ the canonical path space of Z before time 0 instead of $\mathcal{X}_0$. In addition, a past configuration before 0 is denoted by x and the intensity at time 0 is $\phi_i(x)$ instead of $\phi_{i,0}(x_0)$. Moreover, if $x = \left(\{t_n^i\}_{n\in\mathbb{Z}_-}\right)_{i\in\mathbf{I}} \in \mathcal{X}$, then for any $i \in \mathbf{I}$, we denote the point measure associated to index i by

$$dx_s^i = \sum_{m\in\mathbb{Z}^-} \delta_{t_m^i}(ds)$$

in which $\delta_t(.)$ is a Dirac measure at t. Throughout this article, without further mentioning, the integral $\int_a^b$ stands for $\int_{[a,b)}$ with $a, b \in \mathbb{R}$.

Let $x_t = \left(\{t_n^i\}_{n\in\mathbb{Z}}\right)_{i\in\mathbf{I}} \in \mathcal{X}_t$ and denote $x_t^{\leftarrow t} := \left(\{t_n^i - t\}_n\right)_i \in \mathcal{X}$ the shifted configuration. In this paper, if we do not mention otherwise, we always consider time homogeneous point processes, which in our setting can be defined as follows.

Definition 3.3.1. For a given $i \in \mathbf{I}$, a counting process Z^i with stochastic intensity $(\phi_{i,t}(x_t))_{t\in\mathbb{R}}$ is said to be time homogeneous if

$$\phi_{i,t}(x_t) = \phi_i(x_t^{\leftarrow t})$$

for all $t \in \mathbb{R}$ and $x_t \in \mathcal{X}_t$.

Before giving the definition of a Kalikow decomposition, we introduce the definition of a cylindrical function as follows.

Definition 3.3.2. For any neighborhood $v_t \in \mathbf{V}_t$ and $x_t, y_t \in \mathcal{X}_t$, we say $x_t \overset{v_t}{=} y_t$ whenever $x_t = y_t$ in v_t. This means that, for all $i \in \mathbf{I}, n \in \mathbb{Z}$, such that $t_n^i \in x_t$ and $(i, t_n^i) \in v_t$, we have $t_n^i \in y_t$ and vice-versa. A real valued function f is called cylindrical in v_t if $f(x_t) = f(y_t)$ for any $x_t \overset{v_t}{=} y_t$, and we usually stress the dependence in v_t by denoting $f^{v_t}(x_t)$.

In what follows, we give the definition of the Kalikow decomposition for a counting process Z^i with stochastic intensity $\phi_{i,t}(x_t)$.

Definition 3.3.3. We say a time homogeneous process Z^i for some $i \in \mathbf{I}$ admits the Kalikow decomposition with respect to (w.r.t) a neighborhood family $(\mathbf{V}_t)_{t\in\mathbb{R}}$ and a sequence of subspaces $(\mathcal{Y}_t)_{t\in\mathbb{R}}$ of $\mathcal{X}_\infty$, if for all t, the intensity $\phi_{i,t}(x_t)$ admits a convex decomposition for any past configuration $x_t \in \mathcal{X}_t \cap \mathcal{Y}_t$, that is, for any $v_t \in \mathbf{V}_t$ there exists a cylindrical function $\phi_{i,t}^{v_t}(.)$ on v_t taking values in $\mathbb{R}^+$ and a probability density function $\lambda_{i,t}(.)$ such that

$$\forall x_t \in \mathcal{X}_t \cap \mathcal{Y}_t, \qquad \phi_{i,t}(x_t) = \lambda_{i,t}(\emptyset)\phi_{i,t}^{\emptyset} + \sum_{v_t \in \mathbf{V}_t, v_t \neq \emptyset} \lambda_{i,t}(v_t)\phi_{i,t}^{v_t}(x_t) \tag{3.3.1}$$

with $\lambda_{i,t}(\emptyset) + \sum_{v_t \in \mathbf{V}_t, v_t \neq \emptyset} \lambda_{i,t}(v_t) = 1$.

We say process $Z = (Z^i)_{i \in \mathbf{I}}$ satisfies a Kalikow decomposition w.r.t $(\mathbf{V}_t)_{t \in \mathbb{R}}$ and $(\mathcal{Y}_t)_{t \in \mathbb{R}}$ if for all $i \in \mathbf{I}$ each process Z^i satisfies a Kalikow decomposition w.r.t $(\mathbf{V}_t)_{t \in \mathbb{R}}$ and $(\mathcal{Y}_t)_{t \in \mathbb{R}}$. For simplicity, all the conventions of $(\mathcal{X}_t)_{t \in \mathbb{R}}$ will be used for $(\mathcal{Y}_t)_{t \in \mathbb{R}}$, such as $\mathcal{Y}_0$ being replaced by $\mathcal{Y}$, etc.

Remark 5. Note that the function $\lambda_{i,t}(.)$ in Definition 3.3.3 is a deterministic function, that is why this decomposition is unconditional, whereas in [14], $\lambda_{i,t}(.)$ was depending on the dominating Poisson processes (see Introduction). Secondly, we do not restrict ourself to a bounded intensity, which is a notable improvement compared to [19].

In the following subsection, we show that we can translate the decomposition at time 0 to any time t when dealing with time homogeneous processes.

3.3.2 From the decomposition at time 0 to the decomposition at any time t.

In the context of time homogeneous process, for any $t \in \mathbb{R}$, we define

$$\mathcal{Y}_t = \mathcal{Y}^{\to t} = \{x + t | x \in \mathcal{Y}\}.$$

For all $i \in \mathbf{I}$, assume that Equation (3.3.1) is satisfied at time $t = 0$ for any $x \in \mathcal{X} \cap \mathcal{Y}$, we then prove that, for time homogeneous processes, this equation is achieved at any time t, for any $x_t \in \mathcal{X}_t \cap \mathcal{Y}_t$, for a particular choice of $\mathbf{V}_t$. Hence, to show that a counting process Z^i, $i \in \mathbf{I}$ satisfies a Kalikow decomposition, it is then sufficient to write the Kalikow decomposition at time 0 only.

Consider a neighborhood family at time 0, $\mathbf{V}_0$. Take a neighborhood $v \in \mathbf{V}_0$ and for $t \geq 0$, denote $v^{\to t} = \{(i, u + t) : (i, u) \in v\}$, i.e, shift to the right the time component of the neighborhood v by t and define

$$\mathbf{V}_t = \mathbf{V}^{\to t} := \{v^{\to t} : v \in \mathbf{V}_0\}. \tag{3.3.2}$$

Since Equation (3.3.1) is satisfied at time 0 for any $x \in \mathcal{X} \cap \mathcal{Y}$ and since $x_t^{\leftarrow t} \in \mathcal{X} \cap \mathcal{Y}$, this implies that

$$\phi_i(x_t^{\leftarrow t}) = \lambda_{i,0}(\emptyset)\phi_{i,0}^{\emptyset} + \sum_{v \in \mathbf{V}_0, v \neq \emptyset} \lambda_{i,0}(v)\phi_{i,0}^{v}(x_t^{\leftarrow t}).$$

Define the cylindrical function $\varphi_{i,t}^{v^{\to t}}$, that is cylindrical on $v^{\to t}$ such that $\varphi_{i,t}^{v^{\to t}}(x_t) = \phi_{i,0}^{v}(x_t^{\leftarrow t})$ and $\varphi_{i,t}^{\emptyset} := \phi_{i,0}^{\emptyset}$. Moreover, for any nonempty neighborhood v in $\mathbf{V}_0$, we consider $\lambda_{i,t}(v^{\to t}) = \lambda_{i,0}(v)$. Since Z^i is a time homogeneous process, by Definition 3.3.1, we have that

$$\phi_{i,t}(x_t) = \lambda_{i,t}(\emptyset)\varphi_{i,t}^{\emptyset} + \sum_{v^{\to t}\in \mathbf{V}_t, v^{\to t}\neq\emptyset} \lambda_{i,t}(v^{\to t})\varphi_{i,t}^{v^{\to t}}(x_t) \tag{3.3.3}$$

with

$$\lambda_{i,t}(\emptyset) + \sum_{v^{\to t}\in \mathbf{V}_t, v^{\to t}\neq\emptyset} \lambda_{i,t}(v^{\to t}) = 1.$$

This shows that Equation (3.3.1) is satisfied at any time t. Thus, by definition, Z^i admits a Kalikow decomposition w.r.t the neighborhood family $(\mathbf{V}_t)_{t\in\mathbb{R}}$ where $\mathbf{V}_t = \mathbf{V}^{\to t}$. In addition, $(\mathbf{V}_t)_{t\in\mathbb{R}}$ is completely determined once the neighborhood family at time 0, $\mathbf{V}_0$, is determined. From now on, it is sufficient to mention $\mathbf{V}_0$ whenever we speak about the Kalikow decomposition.

3.3.3 About the subspace $\mathcal{Y}$

The role of the subspace $\mathcal{Y}$ is to make the Kalikow decomposition achievable. There are many possible choices for such a subspace, depending on the model under consideration. In this paper, we focus for instance on the choice $\mathcal{Y} = \mathcal{X}^{>\delta}$, the subspace of $\mathcal{X}$ where the distance between any two consecutive possible jumps is greater than δ. More precisely,

$$\mathcal{X}^{>\delta} = \{x = (\{t_n^i\}_{n\in\mathbb{Z}^-})_{i\in\mathbf{I}} \quad \text{such that} \quad \forall n, i \quad t_{n+1}^i - t_n^i > \delta \,\text{and}\, t_0^i \leq 0\}. \tag{3.3.4}$$

Another possible choice of a subspace $\mathcal{Y}$ that we consider guarantees that the intensity is finite, by introducing

$$\mathcal{Y} = \{x \in \mathcal{X}, \forall i \quad \phi_i(x) < \infty\}.$$

In practice, either the process is known, namely, the formula of the intensity is given, and we dispose of theoretical results that guarantee for instance that $x \in \mathcal{Y}$ almost surely, or we handle the decomposition by creating a certain threshold. Before the process reaches this threshold, we are able to achieve the Kalikow decomposition by following one of the methods described in the following section. However, after having reached this threshold we can not say anything any more. This depends of course on the initial condition, and if we start for instance with no point on $(-\infty, 0)$, the intensity is in many cases likely to be finite at least at time 0. To proceed further with a Perfect Simulation algorithm is more tricky since we need to check that the intensity remains finite during all steps of the algorithm. This will be the main object of a future work.

3.4 Main results

3.4.1 The first method

In this section, we present step by step a general method to prove the existence of a Kalikow decomposition for the counting process $Z^i, i \in \mathbf{I}$. We start by discussing the relevant family of neighborhoods.

For any subset $J \subset \mathbf{I}$ we say a process Z^i is locally dependent on a subprocess $Z_J := (Z^j)_{j\in J}$ if the intensity $\phi_i(x)$ is a cylindrical function on $J \times (-\infty, 0)$. Roughly speaking, a process Z^i is locally dependent on a subprocess Z_J if the information of process Z^j with $j \in J$ is compulsory to compute the intensity of process Z^i. For short, we say i is locally dependent on J. A more formal definition of local dependence can be found at [9]. Denote

$$\mathcal{V}_{\cdot\to i} := \{j \in \mathbf{I} \quad \text{such that } i \text{ is locally dependent on } j\} \tag{3.4.1}$$

and $\mathcal{S}^i := \mathcal{V}_{\cdot\to i} \times (-\infty, 0)$.

We denote by $(\mathbf{P}, \leq)$ a countable, ordered set. To simplify notation, in what follows, we consider $\mathbf{P}$ to be $\mathbb{N}$, but depending on the concrete examples, different sets $\mathbf{P}$ will be considered, for example $\mathbb{N} \times \mathbb{N}$, etc.

Define a family of finite space-time neighborhoods at time 0 associated to the dynamic of Z^i:

$$\mathbf{V}^i = \{(v_k^i)_{k\in\mathbf{P}} \subset \mathbf{I} \times (-\infty, 0) \quad \text{such that} \quad \cup_k v_k^i = \mathcal{S}^i\} \tag{3.4.2}$$

with the convention that $v_0^i = \emptyset$.

In order to prove that Z^i admits a Kalikow decomposition w.r.t the neighborhood family $\mathbf{V}^i$ and a convenient subspace $\mathcal{Y}$, we use the following condition. Once this condition is fulfilled, we will show in Proposition 3 below that we can derive the corresponding Kalikow decomposition.

Assumption 1. There exists a non-negative sequence of functions $(\Delta_k^i(x))_{k\in\mathbf{P}}$ which are cylindrical on v_k^i such that

$$\sum_{k=0}^{n} \Delta_k^i(x) \to \phi_i(x)$$

as $n \to \infty$, for every $x \in \mathcal{X} \cap \mathcal{Y}$.

Now, we present a method to obtain a Kalikow decomposition under Assumption 1.

Step 1: For any $k \geq 0$, take $\Delta_k^i(x)$ from Assumption 1, it then guarantees that $\phi_i(x)$ can be written as follows:

$$\phi_i(x) = \Delta_0^i + \sum_{k\geq 1} \Delta_k^i(x)$$

where $\Delta_0^i := \Delta_0^i(x)$ is a constant, that is, it does not depend on x since $v_0^i = \emptyset$. Note that, for $k \geq 1$, by definition, $\Delta_k^i(x)$ is cylindrical on v_k^i and non negative.

Step 2: Define a deterministic nonnegative sequence $(\eta_k^i)_{k\in\mathbf{P}}$ such that

$$\sum_{k\geq 0} \eta_k^i = 1.$$

Obviously, $\phi_i(x)$ can be written as:

$$\phi_i(x) = \eta_0^i \frac{\Delta_0^i}{\eta_0^i} + \sum_{k\geq 1} \eta_k^i \frac{\Delta_k^i(x)}{\eta_k^i}$$

with the convention that $0/0 = 1$. We can set η_k^i to 0 whenever $\Delta_k^i(x)$ equals 0 for all x and even discard the corresponding neighborhood from $\mathbf{V}^i$.

Proposition 3. Consider a point process $Z^i, i \in \mathbf{I}$, with intensity at time 0, $\phi_i(x)$, for any $x \in \mathcal{X}$. For the neighborhood family $\mathbf{V}^i$ defined in (3.4.2), if $\phi_i(x)$ satisfies Assumption 1, then Z^i has the following Kalikow decomposition with respect to $\mathbf{V}^i$ and the subspace $\mathcal{Y}$:

$$\begin{cases} \lambda_i(\emptyset) = \eta_0^i \\ \phi_i^{\emptyset} = \dfrac{\Delta_0^i}{\eta_0^i} \\ \lambda_i(v_k^i) = \eta_k^i \\ \phi_i^{v_k^i}(x) = \dfrac{\Delta_k^i(x)}{\eta_k^i} \end{cases}$$

for any choice of non negative weights $(\eta_k^i)_{k\in\mathbf{P}}$ such that

$$\sum_{k\geq 0} \eta_k^i = 1.$$

Remark 6. From Step 2, it is clear that such a Kalikow decomposition is not unique. However, to perform the Perfect Simulation algorithm (see Section 3.5), $(\eta_k^i)_{k\in\mathbf{P}}$ can not be chosen arbitrarily. These weights need to satisfy several conditions, for example: the stopping condition, see Proposition 7 below, which enables the algorithm to end. On the other hand, for the simulation purposes, these weights should be chosen carefully to avoid the explosion of $\phi_i^{v_k^i}(x)$. In addition, they also influence the mean number of total simulated points of this algorithm, and therefore the efficiency of the algorithm (see Section 3.5). To illustrate this point more clearly, one example is considered in Section 3.5.

3.4.2 Examples of the first method

Linear Hawkes process.

In the following, we consider a linear Hawkes process [12, 13], where the intensity at time 0, $\phi_i(x)$, is given as follows. For any $x \in \mathcal{X}$,

$$\phi_i(x) = \mu_i + \sum_{j\in\mathbf{I}} \int_{-\infty}^{0} h_{ji}(-s)dx_s^j,$$

where $h_{ji}(.)$ measures the local dependence of process Z^i on Z^j and μ_i refers to the spontaneous rate of process Z^i.

We consider the following assumption for the method to work.

Assumption 2. For all $i, j \in \mathbf{I}$, $h_{ji}(.)$ is a non negative function.

Denote ϵ an arbitrary, fixed and positive number. Then we have

$$\phi_i(x) = \mu_i + \sum_{j\in\mathbf{I}} \sum_{n\in\mathbb{N}^*} \int_{-n\epsilon}^{-n\epsilon+\epsilon} h_{ji}(-s)dx_s^j.$$

Furthermore, in this example, we have

$$\mathcal{V}_{\cdot\to i} = \{j \in \mathbf{I} \quad \text{such that} \quad h_{ji} \neq 0\}.$$

Since $\mathbf{I}$ is at most countable, we may enumerate the indices in $\mathbf{I}$ by j_l for $l \in \mathbb{N}^*$. In this example we put $\mathbf{P} = \mathbb{N}^* \times \mathbb{N}^*$, and we consider the neighborhood family $\mathbf{V}^{atom}$ as follows: for $l, n \geq 1$

$$v^i_{(l,n)} := \{j_l\} \times [-n\epsilon, -n\epsilon + \epsilon).$$

We define, for all $l, n \geq 1$:

$$\Delta^i_{(l,n)}(x) := \int_{-n\epsilon}^{-n\epsilon+\epsilon} h_{j_l i}(-s) dx^{j_l}_s \geq 0,$$

and $\Delta^i_{(0,0)} := \Delta^i_{(0,0)}(x) := \mu_i$. Applying the Monotone Convergence theorem, we obtain

$$\Delta^i_{(0,0)} + \sum_{(l,n)=(1,1)}^{(L,N)} \Delta^i_{(l,n)}(x) \to \phi_i(x),$$

when $(L, N) \to (\infty, \infty)$.

Notice that the intensity $\phi_i(x)$ is finite if and only if the series converges. However, what we do here is more general, since we allow the intensity to be infinite.

Thus, Assumption 1 is fulfilled. As a consequence, relying on Proposition 3, we conclude that Z^i has a Kalikow decomposition with respect to $\mathbf{V}^{atom}$ with

$$\begin{cases} \lambda_i(\emptyset) = \eta_{(0,0)} \\ \phi_i^{\emptyset} = \dfrac{\mu_i}{\eta_{(0,0)}} \\ \lambda_i\left(v^i_{(l,n)}\right) = \eta_{(l,n)} \\ \phi_i^{v^i_{(l,n)}}(x) = \dfrac{\int_{-n\epsilon}^{-n\epsilon+\epsilon} h_{j_l i}(-s) dx^{j_l}_s}{\eta_{(l,n)}}, \end{cases}$$

for any choice of non negative weights $\left(\eta^i_{(l,n)}\right)_{(l,n)\in\mathbb{N}^*\times\mathbb{N}^*}$ such that

$$\eta^i_{(0,0)} + \sum_{l,n\geq 1} \eta^i_{(l,n)} = 1.$$

Remark 7. Note that under mild conditions (see for instance [8]) it is well-known that the process exists in a stationary regime. On the other hand, for our purpose it is sufficient to consider the subspace $\mathcal{Y} = \{x \in \mathcal{X} \mid \forall i \quad \phi_i(x) < \infty\}$ such that for all $x \in \mathcal{X} \cap \mathcal{Y}$, $\phi_i(x)$ and $\phi_i^v(x)$ are finite.

Age dependent Hawkes process with hard refractory period.

In this section, we are interested in writing a Kalikow decomposition for Age dependent Hawkes processes with hard refractory period. Up to our knowledge, this process was first introduced in [6] and no Kalikow decomposition has been proved, even in a conditional framework, for this type of process. In our setting, the stochastic intensity of an Age dependent Hawkes process with hard refractory of length $\delta > 0$ can be written as follows. For any $i \in \mathbf{I}$ and $x \in \mathcal{X}$,

$$\phi_i(x) = \psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dx_s^j\right)\mathbb{1}_{a_0^i(x)>\delta} \tag{3.4.3}$$

with the convention that $a_0^i(x) = -L_0^i(x)$ and

$$L_0^i(x) = \sup\{t_k^i \in x \quad \text{such that} \quad t_k^i < 0\} = t_{-1}^i$$

is the last jump before 0 of process Z^i, if it has at least one jump. Otherwise, we put $L_0^i(x) = -\infty$. Again, we have

$$\mathscr{V}_{\cdot\to i} = \{j \in \mathbf{I} \quad \text{such that} \quad h_{ji} \neq 0\} \cup \{i\}.$$

By definition of stochastic intensity (3.4.3), we observe that the distance of any two consecutive jumps have to be larger than δ. This observation leads us to consider the subspace $\mathscr{Y} = \mathscr{X}^{>\delta}$ that is introduced in (3.3.4). To prove a Kalikow decomposition, we consider the following assumptions.

Assumption 3. (i) For all $i, j \in \mathbf{I}$, $h_{ji}(.)$ is a non negative, non increasing L_1 function. Moreover, for every i

$$\sum_{j\in\mathbf{I}} \|h_{ji}\|_{L_1} < \infty.$$

(ii) For every i, $\psi_i(.)$ is an increasing, non negative continuous function.

On the other hand, we build a neighborhood family $\mathbf{V}^{nested}$ by introducing a non decreasing sequence $(V_i(k))_{k\geq 0}$ of finite subsets of $\mathbf{I}$ such that

$$V_i(0) = \emptyset, V_i(1) = \{i\}, V_i(k-1) \subset V_i(k) \quad \text{and} \quad \cup_k V_i(k) = \mathscr{V}_{\cdot\to i} \cup \{i\}. \tag{3.4.4}$$

Consider $\mathbf{P} = \mathbb{N}$ and set $v_k^i = V_i(k) \times [-k\delta, 0)$, then we obtain by construction an increasing, nested neighborhood sequence $(v_k^i)_{k\in\mathbb{N}}$. By applying the methodology of Section 3.4.1, we prove the following Proposition.

Proposition 4. Consider $\mathscr{X}^{>\delta}$ defined as in (3.3.4). For any past configuration $x \in \mathscr{X}$, suppose that the intensity is of the form

$$\phi_i(x) = \psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dx_s^j\right)\mathbb{1}_{a_0^i(x)>\delta}.$$

If Assumption 3 is fulfilled, then the corresponding Kalikow decomposition with respect to $\mathbf{V}^{nested}$ and $\mathscr{X}^{>\delta}$ is for $k \geq 1$

$$\begin{cases} \lambda_i(v_k^i) = \eta_k^i \\ \phi_i^{v_k^i}(x) = \dfrac{\Delta_k^i(x)}{\eta_k^i} \end{cases}$$

with

(i) any positive sequence $(\eta_k^i)_{k\geq 1}$ such that $\sum_{k\geq 1} \eta_k^i = 1$,

(ii)

$$\Delta^i_k(x) = \psi_i\left(\sum_{j\in V_i(k)}\int_{-k\delta}^{0} h_{ji}(-s)dx^j_s\right)\mathbb{1}_{a^i_0(x)>\delta} - \psi_i\left(\sum_{j\in V_i(k-1)}\int_{-k\delta+\delta}^{0} h_{ji}(-s)dx^j_s\right)\mathbb{1}_{a^i_0(x)>\delta}$$

for $k \geq 2$ and $\Delta^1_i(x) = \psi_i(0)\mathbb{1}_{a^i_0(x))>\delta}$.

Remark 8. Note that $\lambda_i(\emptyset)$ does not appear in the above Kalikow decomposition. Amazingly, this does not cause any problems for the Perfect Simulation algorithm (see Section 3.5). We stress that this is one of the main differences with [14], the other one being that the decomposition does not depend on the dominating Poisson processes.

Proof. For any $x \in \mathcal{X}^{>\delta}$, $i \in \mathbf{I}$ and $k \geq 2$, consider

$$\Delta^i_k(x) = \psi_i\left(\sum_{j\in V_i(k)}\int_{-k\delta}^{0} h_{ji}(-s)dx^j_s\right)\mathbb{1}_{a^i_0(x)>\delta} - \psi_i\left(\sum_{j\in V_i(k-1)}\int_{-k\delta+\delta}^{0} h_{ji}(-s)dx^j_s\right)\mathbb{1}_{a^i_0(x)>\delta}. \quad (3.4.5)$$

By Assumption 3, $(\Delta^i_k(x))_{k\in\mathbb{N}}$ is well-defined, non negative and cylindrical on v^i_k. Moreover, we set $\Delta^i_0 = 0$ as well $\eta^i_0 = 0$ and $\Delta^1_i(x) = \psi_i(0)\mathbb{1}_{a^i_0(x))>\delta}$. Let

$$r^{[n]}_i(x) := \sum_{k=1}^{n}\Delta^i_k(x) = \psi_i\left(\sum_{j\in V_i(n)}\int_{-n\delta}^{0} h_{ji}(-s)dx^j_s\right)\mathbb{1}_{a^i_0(x)>\delta}$$

and let us show that $r^{[n]}_i(x) \to \phi_i(x)$ when $n \to \infty$. Consider the inner-term of the parenthesis,

$$\sum_{j\in V_i(n)}\int_{-n\delta}^{0} h_{ji}(-s)dx^j_s = \int_{I\times\mathbb{R}^-} h_{ji}(-s)\mathbb{1}_{(j,s)\in v^i_n}dx^j_s d\kappa_j,$$

where we denote $d\kappa$ the counting measure on the discrete set $\mathbf{I}$.

We have that $\left(h_{ji}(-s)\mathbb{1}_{(j,s)\in v^i_n}\right)_{n\in\mathbb{Z}}$ is non negative and non decreasing sequence in n. In addition, it converges to $h_{ji}(-s)\mathbb{1}_{(j,s)\in\mathcal{V}_{\to i}\times(-\infty,0)}$ as $n \to \infty$. Moreover, since $\psi_i(.)$ is a continuous and increasing function, the Monotone convergence theorem for Lebesgue Stieltjes measures implies that $r^{[n]}_i(x) \to \phi_i(x)$ as $n \to \infty$. As a consequence, Assumption 1 is fulfilled, and by Proposition 3, the conclusion follows. □

Remark 9. Denote $D_i^k(x) = \{z \in \mathcal{X}^{>\delta} : z \overset{v_k^i}{=} x\}$. Clearly,

$$\psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dz_s^j\right)\mathbb{1}_{a_0^i(x)>\delta} = \psi_i\Big(\sum_{j\in V_i(k)}\int_{-k\delta}^{0} h_{ji}(-s)dz_s^j + \\ + \sum_{j\in V_i(k)}\int_{-\infty}^{-k\delta} h_{ji}(-s)dz_s^j + \sum_{j\notin V_i(k)}\int_{-\infty}^{0} h_{ji}(-s)dz_s^j\Big)\mathbb{1}_{a_0^i(x)>\delta}.$$

By Assumption 3, we conclude that,

$$\inf_{z\in D_i^k(x)}\psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dz_s^j\right)\mathbb{1}_{a_0^i(x)>\delta} = \psi_i\left(\sum_{j\in V_i(k)}\int_{-k\delta}^{0} h_{ji}(-s)dx_s^j\right)\mathbb{1}_{a_0^i(x)>\delta}.$$

The equality is achieved when we consider the configuration z having points only inside the neighborhood v_k^i.

Hence, for $k \geq 2$, we observe that

$$\Delta_k^i(x) = \inf_{z\in D_i^k(x)}\psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dz_s^j\right)\mathbb{1}_{a_0^i(x)>\delta} \\ - \inf_{z\in D_i^{k-1}(x)}\psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dz_s^j\right)\mathbb{1}_{a_0^i(x)>\delta}.$$

The above prescription corresponds to the classical method of obtaining a Kalikow decomposition in discrete time, discussed in [10, 14].

3.4.3 Another method for nonlinear Hawkes processes

In this section, we present a second method to prove the existence of a Kalikow decomposition for nonlinear Hawkes processes [4]. In the setting of this section, we suppose that the intensity of the nonlinear Hawkes process $Z^i, i \in \mathbf{I}$, is given by

$$\phi_i(x) = \psi_i\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0} h_{ji}(-s)dx_s^j\right)$$

for any $x \in \mathcal{X}$.

We work under the following assumptions.

Assumption 4. (i) For any $i, j \in \mathbf{I}$, the function $h_{ji}(.)$ is non negative and belongs to L_1. Moreover, for every i, we have

$$\sum_{j\in\mathbf{I}}\|h_{ji}\|_{L_1} < \infty.$$

(ii) For every $i \in \mathbf{I}$, $\psi_i(.)$ is an analytic function on $\mathbb{R}$ with radius of convergence about 0 which is given by K, for some positive K. Moreover, its derivative of order n, $\psi_i^{(n)}(0)$, is non negative for all $n \geq 1$, and $\psi_i(0)$ is non negative as well.

In this section, to develop our series using Taylor expansion, we choose

$$\mathscr{Y}^K = \left\{ x \mid \sup_i \left(\sum_j \int_{-\infty}^0 h_{ji}(-s) dx_s^j \right) < K \right\}.$$

We now describe our method. We fix ϵ an arbitrary positive parameter, and we put $D = \mathbf{I} \times \mathbb{N}$. An index in D is $\alpha = (j, n)$ where $j \in \mathbf{I}$ and $n \in \mathbb{N}$. We put $|\alpha| := |j| + n$. For any $x \in \mathscr{X} \cap \mathscr{Y}^K$, we introduce

$$a_\alpha(x) := \int_{-(n+1)\epsilon}^{-n\epsilon} h_{ji}(-s) dx_s^j.$$

Note that, whenever $x \in \mathscr{X} \cap \mathscr{Y}^K$, we have $a_\alpha(x)$ converges to 0 when $|\alpha| \to \infty$. Furthermore, we have

$$\phi_i(x) = \psi_i\left(\sum_{j\in\mathbf{I}} \sum_{n\in\mathbb{N}} \int_{-(n+1)\epsilon}^{-n\epsilon} h_{ji}(-s) dx_s^j \right) = \psi_i\left(\sum_{j\in\mathbf{I}} \sum_{n\in\mathbb{N}} a_{jn}(x) \right) = \psi_i\left(\sum_{\alpha\in D} a_\alpha(x) \right).$$

We suppose moreover that

Assumption 5. For any $i \in \mathbf{I}$, there exist a deterministic sequence $(\eta^i_{\alpha_k})_{\alpha_k\in D} \in (0,1)$ such that

$$\sum_{n\geq 1} \sum_{\alpha_1,\ldots,\alpha_n\in D} \eta^i_{\alpha_1} \ldots \eta^i_{\alpha_n} < 1.$$

Let us consider $\mathbf{P} = \cup_{n=1}^\infty D^n$ where $D^n := D \times D \times \ldots \times D$ (n times). We construct the neighborhood family $\mathbf{V}^{Taylor}$ by defining for $\alpha_{1:k} = (\alpha_1, \ldots, \alpha_k)$,

$$v^i_{\alpha_{1:k}} = v^i_{(\alpha_1,\ldots,\alpha_k)} := \cup_{l=1}^k w_{\alpha_l}, \tag{3.4.6}$$

where $w_{\alpha_l} := \{j\} \times [-(n+1)\epsilon, n\epsilon)$ if $\alpha_l = (j,n)$, and $v^i_0 := \emptyset$.

Proposition 5. Consider a non linear Hawkes process $Z^i, i \in \mathbf{I}$, with intensity $\phi_i(x)$ given by

$$\phi_i(x) = \psi_i\left(\sum_{j\in\mathbf{I}} \int_{-\infty}^0 h_{ji}(-s) dx_s^j \right).$$

Under Assumption 4 and 5, Z^i has a Kalikow decomposition with respect to the neighborhood family $\mathbf{V}^{Taylor}$ defined by (3.4.6) and $\mathscr{Y}^K$ as follows:

$$\begin{cases} \lambda_i(\emptyset) & = 1 - \sum_{k\geq 1} \sum_{\alpha_1,\ldots,\alpha_k\in D} \eta^i_{\alpha_1} \ldots \eta^i_{\alpha_k} \\ \phi_i^\emptyset & = \dfrac{\psi_i(0)}{\lambda_i(\emptyset)} \\ \lambda_i(v^i_{\alpha_{1:k}}) & = \eta^i_{\alpha_1} \ldots \eta^i_{\alpha_k} \\ \phi_i^{v^i_{\alpha_{1:k}}}(x) & = \dfrac{\psi_i^{(k)}(0)}{k!} \dfrac{a_{\alpha_1}(x)}{\eta^i_{\alpha_1}} \ldots \dfrac{a_{\alpha_k}(x)}{\eta^i_{\alpha_k}}, \end{cases}$$

for any weights $(\eta^i_{\alpha_k})_{\alpha_k\in D}$ satisfying Assumption 5 and for $\alpha=(j,n)$, a_α defined by

$$a_\alpha(x) := \int_{-(n+1)\epsilon}^{-n\epsilon} h_{ji}(-s)dx^j_s.$$

Proof. From Assumption 4, (ii), and noticing that for any $x\in\mathcal{X}\cap\mathcal{Y}^K$, we have $0\le\sum_{j\in\mathbf{I}}\sum_{n\in\mathbb{N}}a_{jn}(x)=\sum_{\alpha\in D}a_\alpha(x)<K$, we have a Taylor expansion of $\phi_i(x)$:

$$\begin{aligned}\phi_i(x)&=\psi_i\left(\sum_{j\in\mathbf{I}}\sum_{n\in\mathbb{N}}a_{jn}(x)\right)=\psi_i\left(\sum_{\alpha\in D}a_\alpha(x)\right)\\&=\psi_i(0)+\sum_{k\ge1}\frac{\psi_i^{(k)}(0)}{k!}\left(\sum_{\alpha=(j,n)\in D}a_\alpha(x)\right)^k\\&=\psi_i(0)+\sum_{k\ge1}\sum_{\alpha_1\dots\alpha_k\in D}\frac{\psi_i^{(k)}(0)}{k!}a_{\alpha_1}(x)\dots a_{\alpha_k}(x).\end{aligned}$$

Thus, to obtain the Kalikow decomposition, take $(\eta^i_{\alpha_k})_{\alpha_k\in D}$ from Assumption 5, then the conclusion follows. $\square$

Remark 10. The parameter $(\eta^i_{\alpha_k})_{\alpha_k\in D}$ plays the same role here as $(\eta^i_k)_{k\in\mathbb{N}}$ in the previous section. Again, the choice of this parameter is discussed in Section 3.5.

In the following, we give several examples in which we can apply the second method.

3.4.4 Examples of second method

In this section, we always consider a deterministic sequence $(\eta^i_{\alpha_k})_{\alpha_k\in D}$ that satisfies Assumption 5.

Example 1. Exponential Hawkes process with $\psi_i(.) := \exp(.)$. In this case we have $K=\infty$, $\psi_i(0)=1$ and $\psi_i^{(k)}(0)=1$ for $k\ge1$. Therefore,

$$\begin{aligned}\phi_i(x)&=\exp\left(\sum_{j\in\mathbf{I}}\int_{-\infty}^{0}h_{ji}(-s)dx^j_s\right)\\&=1+\sum_{k\ge1}\sum_{\alpha_1\dots\alpha_k\in D}\frac{1}{k!}a_{\alpha_1}(x)\dots a_{\alpha_k}(x).\end{aligned}$$

Then, the Kalikow decomposition of Z^i with respect to the neighborhood family $\mathbf{V}^{Taylor}$ in (3.4.6) and $\mathcal{Y}^\infty$ is given by

$$\begin{cases}\lambda_i(\emptyset) &= 1-\sum_{k\ge1}\sum_{\alpha_1,\dots,\alpha_k\in D}\eta^i_{\alpha_1}\dots\eta^i_{\alpha_k}\\ \phi_i(\emptyset) &= \dfrac{1}{\lambda_i(\emptyset)}\\ \lambda_i(v^i_{\alpha_{1:k}}) &= \eta^i_{\alpha_1}\dots\eta^i_{\alpha_k}\\ \phi_i^{v^i_{\alpha_{1:k}}}(x) &= \dfrac{1}{k!}\times\dfrac{a_{\alpha_1}(x)}{\eta^i_{\alpha_1}}\dots\dfrac{a_{\alpha_k}(x)}{\eta^i_{\alpha_k}}.\end{cases}$$

Example 2. We consider the non-linear Hawkes process with $\psi_i(u) = ch(u)$ where $ch(u) = \dfrac{e^u + e^{-u}}{2}$. Then $K = \infty$, $\psi_i(0) = 1$, and for $k \geq 1$, we have $\psi_i^{(2k)}(0) = 0$ and $\psi_i^{(2k-1)}(0) = \dfrac{1}{2}$.

Therefore, Z^i has the following Kalikow decomposition with respect to the neighborhood family $\mathbf{V}^{Taylor}$ in (3.4.6) and $\mathscr{Y}^\infty$ is given by

$$\begin{cases} \lambda_i(\emptyset) & = 1 - \sum_{k\geq 1} \sum_{\alpha_1,\ldots,\alpha_{2k-1}} \eta^i_{\alpha_1} \cdots \eta^i_{\alpha_{2k-1}} \\ \phi_i(\emptyset) & = \dfrac{1}{\lambda_i(\emptyset)} \\ \lambda_i(v^i_{\alpha_{1:2k-1}}) & = \eta^i_{\alpha_1} \cdots \eta^i_{\alpha_{2k-1}} \\ \phi_i^{v^i_{\alpha_{1:2k-1}}}(x) & = \dfrac{1}{2(2k-1)!} \times \dfrac{a_{\alpha_1}(x)}{\eta^i_{\alpha_1}} \cdots \dfrac{a_{\alpha_{2k-1}}(x)}{\eta^i_{\alpha_{2k-1}}}. \end{cases}$$

Remark 11. It is well-known that the Exponential Hawkes process may explode in finite time with probability non null [5] (that is, produce an infinite number of jumps within a finite time interval). To perform the Perfect Simulation, in this case, we shall consider the process having empty past before time 0, in other words, the process starting from 0. However, the difficulty lies on the fact that we need to check that the process stays in $\mathscr{Y}^\infty$. To avoid this difficulty, we shall not consider the Exponential Hawkes process or other potentially explosive processes in the following section. The discussion of potentially explosive process will be the main object of upcoming work, as we mentioned earlier.

3.5 Modified Perfect Simulation algorithm

In the following, we present a Perfect Simulation algorithm that simulates the process Z^i on an interval $[0, tmax]$, for some fixed $tmax > 0$ in the stationary regime. Our algorithm is a modification of the method described in [19]. The procedure consists of backward and forward steps. In the backward steps, thanks to the Kalikow decomposition, we create a set of ancestors, which is a list of all the points that might influence the point under consideration. On the other hand, in the forward steps, where we go forward in time, by using the thinning method [17] we give the decision (keep or reject) to each visited point based on its neighborhood until the state of all considered points is decided. Further discussion can be found in [10, 14, 19].

For this algorithm to work, we introduce a subspace $\mathscr{Y}$ as follows:

$$\mathscr{Y} = \{x \in \mathscr{X} \quad \text{such that} \quad \forall v, i : \quad \phi_i^v(x) \leq \Gamma_i\},$$

where Γ_i is a positive constant. We assume that the process generated by the algorithm stays in $\mathscr{Y}$ almost surely. For example, in the case of age dependent Hawkes processes in Section 3.4.2, we show in Proposition 9 below that $\Delta^i_k(x)$ are bounded. Thus, with an adequate choice of weights $(\lambda_i(.))_{i\in\mathbf{I}}$ and $(\Gamma_i)_{i\in\mathbf{I}}$, the algorithm remains in $\mathscr{Y}$ almost surely. Moreover, for any $x \in \mathscr{Y}$, we have

$$\phi_i(x) = \lambda_i(\emptyset)\phi_i^\emptyset + \sum_{\emptyset \neq v \in \mathbf{V}^i} \lambda_i(v)\phi_i^v(x) \leq \Gamma_i,$$

which means that the intensity is bounded.

Furthermore, for any t and $x_t \in \mathcal{Y}_t$, by the time homogeneity assumption,

$$\phi_{i,t}(x_t) = \phi_i(x_t^{\leftarrow t}) \leq \Gamma_i.$$

3.5.1 Backward procedure

Initial step. Fix i, set the initial time to be 0.

Step 1. Move to the first possible jump T of Z^i after 0 by taking

$$T \leftarrow 0 + Exp(\Gamma_i).$$

Step 2. Recall that $\mathbf{V}^i$ is the neighborhood family associated to index i. Independently of anything else, pick a random neighborhood $V_{i,T}$ of (i,T) according to the distribution $(\lambda_i(v))_{v\in\mathbf{V}^i}$, which corresponds to the Kalikow decomposition of intensity $\phi_{i,T}(x_T)$ in (3.3.1) where $x_T \in \mathcal{Y}_T$.

More precisely, this means that we attach to (i,T) a random variable $V_{i,T}$ with values in $(\mathbf{V}^i)^{\rightarrow T}$. For any $v \in \mathbf{V}^i$,

$$\mathbb{P}(V_{i,T} = v^{\rightarrow T}) = \lambda_i(v).$$

Assume that $V_{i,T} = v^{\rightarrow T}$ and define the projection to the second coordinate of v by

$$\pi_j(v) := \{t \in \mathbb{R} | (j,t) \in v\},$$

for any $j \in \mathbf{I}$. Notice that if for some j, $(j,t) \notin v$ for all t, then $\pi_j(v) = \emptyset$.

Simulate Poisson processes in $v^{\rightarrow T}$, that is for each $j \in \mathbf{I}$, we simulate a Poisson process Π_j with intensity Γ_j on $\pi_j(v^{\rightarrow T})$. Put these points in $\mathscr{C}^1_{i,T}$, call it the first set of ancestors of (i,T):

$$\mathscr{C}^1_{i,T} = \bigcup_{j\in\mathbf{I}} \left\{(j,t) | t \in \Pi_j\left(\pi_j(v^{\rightarrow T})\right)\right\}.$$

Step 3. Recursively, we define the n^{th} set of ancestors of (i,T),

$$\mathscr{C}^n_{i,T} = \bigcup_{(j,s)\in\mathscr{C}^{n-1}_{i,T}} \mathscr{C}^1_{j,s} \setminus \left(\mathscr{C}^1_{i,T} \cup \ldots \cup \mathscr{C}^{n-1}_{i,T}\right),$$

and for each $(j,t') \in \mathscr{C}^n_{i,T}$, we perform Step 1 and Step 2.

The backward scheme stops when $\mathscr{C}^n_{i,T} = \emptyset$. We denote

$$N_{i,T} = \inf\{n : \mathscr{C}^n_{i,T} = \emptyset\}.$$

The genealogy of (i,T) is given by

$$\mathscr{C}_{i,T} = \cup_{k=1}^{N_{i,T}} \mathscr{C}^k_{i,T}.$$

Remark 12. Let us emphasize that $\lambda_i(\emptyset)$ does not need to be strictly positive in the present Kalikow decomposition. This means that there is a chance that at every step of the Backward steps, we need to simulate a Poisson process in a non empty neighborhood. However, if there is no point simulated in these intervals, then we do nothing in the next step. Hence, the Backwards steps end. This is one of the main advantage of this Kalikow decomposition with respect to [10, 14].

Remark 13. Notice that, comparing to the original Backward procedure [10, 14], here, we start by simulating a point in the "future" by adding an exponential random variable to the initial time. Unfortunately, due to this additional step, the algorithm always requires that the intensity is bounded in $\mathcal{Y}$. This is one of the main drawback of this new algorithm.

3.5.2 Forward procedure

We now attach to each point in $\mathcal{C}_{i,T}$ a random variable χ whose value is either 0 or 1, where 1 means that this point is accepted.

For any point (i,t) in the Backward steps, we know its neighborhood which contains all the points that might influence the state of (i,t), but we do not know yet the state χ of the points in the neighborhood.

We start with the point $(j,s) \in \mathcal{C}_{i,T}$ which is the smallest in time, so that its associated neighborhood is either empty ($v = \emptyset$) or non empty but without any point of the Poisson process in it.

Step 1. Assign $\chi_{j,s}$ by Bernoulli variable with parameter $\dfrac{\phi_{j,s}^{V_{j,s}}(x_s)}{\Gamma_j}$ where $V_{j,s}$ is the neighborhood of (j,s) . If $\chi_{j,s} = 1$, this means we accept this point.

Step 2. Move to the next point of $\mathcal{C}_{i,T}$ in increasing time order. Repeat Step 1:2 until $\chi_{i,T}$ is determined.

Update step. Update the starting time of the initial step by T. Repeat the Backward and Forward procedures until the starting time is greater than $tmax$.

Remark 14. When implementing the algorithm on a computer, it is worth noticing that, whenever we simulate in an interval that intersects with previously simulated intervals, we will not simulate in the intersecting parts. In particular, no additional points is simulated in $\{i\} \times (0,T)$ where (i,T) is the first simulated point.

3.5.3 Do we construct the right intensity?

For any $i \in \mathbf{I}$, we denote the arrival times of Π_i in Step 2 of the Backward steps by $(\tau_n^i)_{n\in\mathbb{Z}}$, with τ_1^i being the first positive time. Indeed even if we have simulated it on the randomly generated neighborhood, since the intensity is always the same, we can assume that all these points come only from one single Poisson process.

As in Step 1 of the Forward steps, we attach to each point of Π_i a stochastic mark χ given by,

$$\chi_n^i = \begin{cases} 1 & \text{if} \quad \tau_n^i \quad \text{is accepted,} \\ 0 & otherwise. \end{cases}$$

In addition, for any $i \in \mathbf{I}$, define $\mathscr{Z}^i = (\tau_n^i, \chi_n^i)_{n\in\mathbb{Z}}$ an E-marked point process with $E = \{0; 1\}$. In particular, following the notation in Chapter VIII of [3], for any $i \in \mathbf{I}$, let

$$d\mathscr{Z}_t^i(mark) = \sum_{n\in\mathbb{Z}} \mathbb{1}_{\chi_n^i = mark}\delta_{\tau_n^i}(dt) \quad \text{for} \quad mark \in E,$$
$$\mathscr{F}_{t^-}^{\mathscr{Z}} = \bigvee_{i\in\mathbf{I}} \sigma(\mathscr{Z}_s^i(0), \mathscr{Z}_s^i(1); s < t) \quad \text{and} \quad \mathscr{F}_{t^-}^{\mathscr{Z}(1)} = \bigvee_{i\in\mathbf{I}} \sigma(\mathscr{Z}_s^i(1); s < t).$$

Moreover note that $(\mathscr{Z}_t^i(1))_{t\in\mathbb{R}}$ is the counting process associated to the accepted points of the algorithm. With these notations, we can prove the following result. This proof is already done in [19] but we add it here for sake of completeness.

Proposition 6. If we suppose that the process $(\mathscr{Z}_t^i(1))_{t\in\mathbb{R}}$ stays in $(\mathscr{Y}_t)_{t\in\mathbb{R}}$ almost surely then it admits $\phi_{i,t}(x_t)$ as $\mathscr{F}_{t^-}^{\mathscr{Z}(1)}$-predictable intensity.

Proof. Take C_t a non negative predictable process with respect to $\mathscr{F}_t^{\mathscr{Z}^i(1)}$, that is $\mathscr{F}_{t-}^{\mathscr{Z}(1)}$ measurable and therefore $\mathscr{F}_{t-}^{\mathscr{Z}}$ measurable . We have, for any $i \in \mathbf{I}$,

$$\mathbb{E}\left(\int_0^\infty C_t d\mathscr{Z}_t^i(1)\right) = \sum_{n=1}^\infty \mathbb{E}\Big(C_{\tau_n^i} \mathbb{1}_{\chi_n^i=1}\Big).$$

Note that by Theorem T35 at Appendix A1 of [3], any point T should be understood as a stopping time, and that by Theorem T30 at Appendix A2 of [3],

$$\mathscr{F}_{T-}^{\mathscr{Z}} = \bigvee_{j\in\mathbf{I}} \sigma\{\tau_m^j, \chi_m^j \text{ such that } \tau_m^j < T, T\}.$$

Therefore,

$$\mathbb{E}\left(\int_0^\infty C_t d\mathscr{Z}_t^i(1)\right) = \sum_{n=1}^\infty \mathbb{E}\Big(C_{\tau_n^i}\mathbb{E}(\mathbb{1}_{\chi_n^i=1}|\mathscr{F}_{\tau_n^i-}^{\mathscr{Z}}, V_n^i)\Big) = \sum_{n=1}^\infty \mathbb{E}\left(C_{\tau_n^i}\frac{\phi_{i,\tau_n^i}^{V_n^i}(x_{\tau_n^i})}{\Gamma_i}\right),$$

where we denote V_n^i for the neighborhood of τ_n^i.

Let us now integrate with respect to the choice V_n^i, which is independent of anything else.

$$\begin{aligned}\mathbb{E}\left(\int_0^\infty C_t d\mathscr{Z}_t^i(1)\right) &= \sum_{n=1}^\infty \mathbb{E}\left(C_{\tau_n^i}\frac{\lambda_i(\emptyset)\phi_{i,\tau_n^i}^{\emptyset} + \sum_{v\in\mathbf{V}^{\to\tau_n^i}, v\neq\emptyset} \lambda_i(v)\times\phi_{i,\tau_n^i}^{v}(x_{\tau_n^i})}{\Gamma_i}\right)\\ &= \mathbb{E}\left(\int_0^\infty C_t\frac{\phi_{i,t}(x_t)}{\Gamma_i}d\Pi_i(t)\right).\end{aligned}$$

Since Π_i is a Poisson process with respect to $\mathscr{F}_t^{\mathscr{Z}}$ with intensity Γ_i, and $C_t\frac{\phi_{i,t}(x_t)}{\Gamma_i}$ is $\mathscr{F}_{t-}^{\mathscr{Z}}$ measurable, we finally have that

$$\mathbb{E}\left(\int_0^\infty C_t d\mathcal{Z}_t^i(1)\right) = \mathbb{E}\left(\int_0^\infty C_t \phi_{i,t}(x_t) dt\right),$$

which ends the proof. □

3.5.4 Why does the Backward steps end?

We construct a tree with root (i, T). For each point $(j_{T'}, T')$ in the tree, the points which are simulated in $V_{j_{T'}, T'}$ (Step 2 of the Backward steps) define the children of $(j_{T'}, T')$ in the tree. This forms the tree $\tilde{\mathcal{T}}$.

Let us now build a tree $\tilde{\mathcal{C}}$ with root (i, T) (that includes the previous tree) by mimicking the procedure in the Backward steps, except that we simulate – independently on anything else – on the whole neighborhood even if it has a part that intersects with previous neighborhoods (if they exist) (Step 3 of the Backward steps). By doing so, we make the number of children larger but at each node, they are independent of anything else.

If the tree $\tilde{\mathcal{C}}$ goes extinct, then so does the tree $\tilde{\mathcal{T}}$, and the backward part of the algorithm terminates.

To formulate a sufficient criterion that implies the almost sure extinction of $\tilde{\mathcal{C}}$, let us denote the product measure P on $\mathbf{I} \times \mathbb{R}$, defined on the product sets that generate the Borel subsets of $\mathbf{I} \times \mathbb{R}$, as follows.

$$P(J \times A) := \sum_{j \in J} \Gamma_j \, \mu(A)$$

for any $J \subset \mathbf{I}$, where A is a Borel subset of $\mathbb{R}$ and where μ is the Lebesgue measure. The following proposition is already proved in [19], but without precisely defining P.

Proposition 7. If

$$\sup_{i \in \mathbf{I}} \sum_{k \geq 1} P(v_k^i) \lambda_i(v_k^i) < 1 \tag{3.5.1}$$

then the Backward steps in the Perfect Simulation algorithm terminate almost surely in finite time.

Remark 15. For any neighborhood v, we have

$$\sum_{j \in v} \mathbb{E}\big(\Pi_j\big(\pi_j(v)\big)\big) = P(v).$$

This implies that $\sum_{k \geq 1} P(v_k^i) \lambda_i(v_k^i)$ is the mean number of children issued from one point of type i. Then, the condition (3.5.1) says that the number children in each step should be less than one in average, for the tree to go extinct almost surely. That is a very classical result in Branching process [1].

3.5.5 The complexity of the algorithm

In this section, we study the effect of $(\lambda_i(v_k^i))_{i,k}$ on the number of points simulated by our algorithm. Until the end of this section, we suppose that

Assumption 6. The index set $\mathbf{I}$ is finite, namely

$$|\mathbf{I}| = N < \infty.$$

Let us mention several notations that will be useful in the sequel. We denote e_i the i-th unit vector of $\mathbb{R}^N$, $\mathbf{1}$ is the vector $(1, 1, \dots, 1)^T$ and μ always stands for the Lebesgue measure. Finally, by a positive vector, we mean that all its components are positive.

Coming back to our problem, since the tree $\tilde{\mathscr{C}}$ dominates $\tilde{\mathscr{T}}$, to give an upper bound of the number of points of $\tilde{\mathscr{T}}$, it is sufficient to study the number of points of $\tilde{\mathscr{C}}$. Recall that the root of the tree $\tilde{\mathscr{C}}$ is of type $i \in \mathbf{I}$. For $n \geq 1$, write $K_i(n)$ for the vector containing the numbers of ancestors in the n^{th} set of ancestors of a single point of index i. We consider

$$K_i(n) = \begin{pmatrix} \text{number of ancestors in the} & n^{th} & \text{set of ancestors of type} & 1 \\ \text{number of ancestors in the} & n^{th} & \text{set of ancestors of type} & 2 \\ & \dots & & \\ \text{number of ancestors in the} & n^{th} & \text{set of ancestors of type} & N \end{pmatrix},$$

with the convention that $K_i(0) = e_i$ for every i. For every $j \in \mathbf{I}$, denote $X_i^j := K_i^j(1)$ the number of ancestors of type j in the first set of ancestors with initial point of type i. Moreover, as we discussed earlier, X_i^j is the cardinal of the points that a Poisson process of intensity Γ_j puts on $\pi_j(V_i)$, where V_i is the random neighborhood from a point of type i. In other words, if we denote $\mathscr{P}$ the Poisson distribution, conditioning on $V_i = v$, we have

$$X_i^j \sim \mathscr{P}\big(\Gamma_j \mu(\pi_j(v))\big).$$

In addition, we denote $X_i = (X_i^1, X_i^2, \dots, X_i^N)^T$ and for any $\theta \in \mathbb{R}^N$, we consider the log-Laplace transform of X_i:

$$\phi_i(\theta) := \log \mathbb{E}_i\left(e^{\theta^T X_i}\right)$$

where we denote $\mathbb{P}_i$ the law of a random tree $\tilde{C}$, whose root is of type i and $\mathbb{E}_i$ the corresponding expectation.

For $n \geq 2$, given the population in the first set of ancestors X_i, we have the following relationship between the $(n-1)^{th}$ set of ancestors and the n^{th} set of ancestors in $\tilde{C}$,

$$K_i(n) = \sum_{j \in \mathbf{I}} \sum_{p=1}^{X_i^j} K_j^{(p)}(n-1),$$

where $K_j^{(p)}(n-1)$ is the vector of the number of ancestors in the $(n-1)^{th}$ set of ancestors issued from the p-th point of type j in the first generation. In addition, for $p = 1, \dots, X_i^j$, we have that the $K_j^{(p)}(n-1)$'s are independent copies of $K_j(n-1)$. Note that this equation is trivial for $n = 1$.

Moreover, we consider

$$W_i(n) = \sum_{k=0}^{n} K_i(k)$$

the total number of ancestors in the first n set of ancestors.

The log Laplace transform associated to the random vector $W_i(n)$ is given by:

$$\Phi_i^{(n)}(\theta) := \log \mathbb{E}_i\left(e^{\theta^T W_i(n)}\right).$$

Finally we put $\Phi^{(n)}(\theta) = (\Phi_1^{(n)}(\theta), \dots, \Phi_N^{(n)}(\theta))^T$.

Denote the derivative of $\phi(.)$ at 0 by the matrix $M = D\phi(0)$. Now, let us take a closer look to the function $\phi_i(\theta)$,

$$\phi_i(\theta) = \log \mathbb{E}_i\left(e^{\theta^T X_i}\right) = \log \mathbb{E}_i\left(\prod_{j=1}^{N} e^{\theta_j X_i^j}\right).$$

Therefore the gradient matrix M satisfies $M_{ji} = \mathbb{E}_i(X_i^j)$ for $i, j \in \mathbf{I}$. Recall that, conditioning on $V_i = v$, $X_i^j \sim \mathscr{P}\big(\Gamma_j \mu(\pi_j(v))\big)$, therefore we have

$$M_{ji} = \sum_k \Gamma_j \mu\big(\pi_j(v_k^i)\big) \lambda_i(v_k^i). \tag{3.5.2}$$

Introduce $W_i(\infty) := \lim_{n\to\infty} W_i(n)$, if it exists, and let $W_i = \mathbf{1}^T W_i(\infty)$ be the total number of points of $\tilde{C}$. The log-Laplace transform of $W_i(\infty)$ is given by

$$\Phi_i(\theta) := \log \mathbb{E}_i\left(e^{\theta^T W_i(\infty)}\right),$$

and we write again $\Phi(\theta) = (\Phi_1(\theta), \Phi_2(\theta), \dots, \Phi_N(\theta))^T$.

With this approach, we can prove the following exponential inequality. This result is inspired by Lemma 1 of [21]. Define $\|.\|_1$ a 1− norm on $\mathbb{R}^N$ and the ball with respect to this norm on $\mathbb{R}^N$ by $B(.)$.

Proposition 8. Grant Assumption 6 and assume that the weights $(\lambda_i(.))_{i\in\mathbf{I}}$ satisfy the following condition:

$$\sup_{i\in\mathbf{I}} \sum_{k\geq 1} P(v_k^i)\lambda_i(v_k^i) < 1.$$

Suppose moreover that there exists a positive number r that depends on the matrix M such that, for all positive vectors θ belonging to $B(0, r)$, we have

$$\sup_i \phi_i(\theta) = \sup_i \sum_j \log\left(\sum_k \lambda_i(v_k^i) \exp\left[(e^{\theta_j} - 1)\Gamma_j \mu\big(\pi_j(v_k^i)\big)\right]\right) < \infty.$$

Then for all $\theta \in B(0, r)$, we conclude that $\Phi_i(\theta) < \infty$ and moreover

$$\Phi(\theta) = \theta + \phi(\Phi(\theta)).$$

In particular, for $\vartheta \in \mathbb{R}_+$ such that $\theta = \vartheta\mathbf{1} \in B(0, r)$, there exists a constant c_0 that depends on M and i such that

$$\mathbb{P}\big(W_i > \mathbb{E}(W_i) + x\big) \leq c_0 e^{-\vartheta x} \tag{3.5.3}$$

for all $x > 0$.

Remark 16. From Chapter V of [1] , we have

$$\mathbb{E}\big(K_i(n)\big) = M^n \mathbb{E}\big(K_i(0)\big) = M^n e_i.$$

Therefore, the total number of points simulated of the tree $\tilde{C}$ with initial point of type i is

$$\mathbb{E}\big(W_i(n)\big) = \mathbb{E}\left(\sum_{k=0}^{n} K_i(k)\right) = \left(\sum_{k=0}^{n} M^k\right) e_i.$$

It leads us to conclude that,

$$\mathbb{E}(W_i) = \mathbf{1}^T\left(\sum_{k=0}^{\infty} M^k\right) e_i,$$

where the matrix M is defined in (3.5.2).

To conclude, if we think the complexity of the algorithm in terms of the number of simulated points, W_i is a good bound, and by (3.5.3) we see that we can grasp the complexity of the algorithm by studying (3.5.3).

Proof. First, we prove that $\phi_i(\theta)$ is well defined for all $\theta \in B(0, r)$. Indeed, since $(X_i^j)_{j=1,\ldots,N}$ are independent, we have

$$\begin{aligned}\phi_i(\theta) &= \log \mathbb{E}_i\left(\prod_j \exp\left(\theta_j X_i^j\right)\right)\\ &= \sum_j \log \mathbb{E}_i\left(\exp\left(\theta_j X_i^j\right)\right).\end{aligned}$$

Moreover, by conditioning on the first random neighborhood of type i, V_i, we have

$$X_i^j | V_i = v_k^i \sim \mathscr{P}\big(\Gamma_j \mu(\pi_j(v_k^i))\big).$$

Therefore, we conclude that

$$\mathbb{E}_i\left(\exp\left(\theta_j X_i^j\right) | V_i = v_k^i\right) = \exp\left[(e^{\theta_j} - 1)\Gamma_j \mu\big(\pi_j(v_k^i)\big)\right].$$

Finally, we obtain that

$$\phi_i(\theta) = \sum_j \log\left(\sum_k \lambda_i(v_k^i) \exp\left[(e^{\theta_j} - 1)\Gamma_j \mu\big(\pi_j(v_k^i)\big)\right]\right) < \infty.$$

In the following, we prove that $\Phi_i^{(n)}(\theta)$ satisfies the following recursion

$$\Phi_i^{(n)}(\theta) = \theta^T K_i(0) + \phi_i(\Phi^{(n-1)}(\theta)).$$

Indeed, by definition of $W_i(n)$ we have

$$\mathbb{E}_i\left(e^{\theta^T W_i(n)}\right) = e^{\theta^T K_i(0)} \mathbb{E}_i\left(e^{\theta^T \sum_{k=1}^n K_i(k)}\right).$$

In addition, from the definition of $K_i(k)$ we obtain,

$$\begin{aligned}\mathbb{E}_i\left(e^{\theta^T\sum_{k=1}^n K_i(k)}\right) &= \mathbb{E}_i\left(e^{\theta^T\sum_{k=1}^n\sum_{j=1}^N\sum_{p=1}^{X_i^j}K_j^{(p)}(k-1)}\right)\\ &= \mathbb{E}_i\left(\prod_{j=1}^N e^{\theta^T\sum_{k=1}^n\sum_{p=1}^{X_i^j}K_j^{(p)}(k-1)}\right)\\ &= \mathbb{E}_i\left[\mathbb{E}\left(\prod_{j=1}^N e^{\theta^T\sum_{k=1}^n\sum_{p=1}^{X_i^j}K_j^{(p)}(k-1)}|X_i\right)\right].\end{aligned}$$

Since conditioning on X_i, $K_j^{(p)}(k-1)$'s are independent, we have that

$$\begin{aligned}\mathbb{E}_i\left(e^{\theta^T\sum_{k=1}^n K_i(k)}\right) &= \mathbb{E}_i\left[\prod_{j=1}^N\mathbb{E}\left(e^{\theta^T\sum_{k=1}^n\sum_{p=1}^{X_i^j}K_j^{(p)}(k-1)}|X_i^j\right)\right]\\ &= \mathbb{E}_i\left[\prod_{j=1}^N\prod_{p=1}^{X_i^j}\mathbb{E}\left(e^{\theta^T\sum_{k=1}^n K_j^{(p)}(k-1)}\mid X_i^j\right)\right]\\ &= \mathbb{E}_i\left[\prod_{j=1}^N\prod_{p=1}^{X_i^j}\mathbb{E}\left(e^{\theta^T\sum_{k=1}^n K_j^{(p)}(k-1)}\right)\right]\\ &= \mathbb{E}_i\left[\prod_{j=1}^N\left(\mathbb{E}\left(e^{\theta^T W_j(n-1)}\right)\right)^{X_i^j}\right]\\ &= \mathbb{E}_i\left[\prod_{j=1}^N e^{\Phi_j^{(n-1)}(\theta)X_i^j}\right] = \mathbb{E}_i\left[e^{\Phi^{(n-1)}(\theta)^T X_i}\right] = e^{\phi_i(\Phi^{(n-1)}(\theta))}.\end{aligned}$$

Finally, we conclude that

$$\Phi_i^{(n)}(\theta) = \theta^T K_i(0) + \phi_i(\Phi^{(n-1)}(\theta)).$$

This equation holds for every i, therefore we have

$$\Phi^{(n)}(\theta) = \theta + \phi(\Phi^{(n-1)}(\theta)). \tag{3.5.4}$$

Let us consider the column sums of the matrix M:

$$\sum_{j\in\mathbf{I}} M_{ji} = \sum_{j\in\mathbf{I}}\mathbb{E}_i(X_i^j) = \sum_{k\geq 1} P(v_k^i)\lambda_i(v_k^i).$$

Hence, $\|M\|_1 = \sup_{\|x\|_1\leq 1}\{\|Mx\|_1\} = \sup_i\sum_{j\in\mathbf{I}}|M_{ji}| < 1$ where $\|.\|_1$ is the induced norm for matrix on $\mathbb{R}^{N\times N}$. Therefore, $\|D\phi(0)\|_1 \leq C < 1$. Moreover the norm is continuous and $D\phi(s)$ is likewise, there is a $r > 0$ such that, for $||s||_1 \leq r$,

$$\|D\phi(s)\|_1 \leq C < 1.$$

Hence, $\phi(s)$ is Lipschitz continuous in the ball $B(0,r)$ and moreover $\phi(0)=0$, which implies that

$$\|\phi(s)\|_1 \leq C\|s\|_1$$

for $\|s\|_1 \leq r$.

Moreover, take θ such that $\frac{\|\theta\|_1}{1-C} \leq r$, hence $\|\theta\|_1 \leq r$. By induction we can show that

$$\|\Phi^{(n)}(\theta)\|_1 \leq \|\theta\|_1(1+C+\ldots+C^n) \leq r < \infty.$$

In addition, $(W_i(n))_{n\geq 1}$ is a positive, increasing vector sequence. Here, for any $u, v \in \mathbb{R}^N$, we say $u \geq v$ if $(u-v)$ is a positive vector.

For any θ such that $\frac{\|\theta\|_1}{1-C} \leq r$, by using the theorem of monotone convergence, we have

$$\Phi^{(n)}(\theta) \to \Phi(\theta)$$

when $n \to \infty$. Moreover, $\|\Phi(\theta)\|_1 < \infty$, and passing to the limit $n \to \infty$ in equation (3.5.4), we conclude that

$$\Phi(\theta) = \theta + \phi(\Phi(\theta)).$$

In particular, choosing $\theta = \vartheta\mathbf{1} \in B(0,r)$ with $\vartheta \in \mathbb{R}$, we have

$$\mathbb{E}_i\left(e^{\vartheta W_i}\right) < \infty,$$

where $W_i = \mathbf{1}^T W_i(\infty)$ is the total number of point of $\tilde{C}$.

The last point is concluded by using Markov's inequality, that ends the proof.

□

3.5.6 Efficiency of the algorithm and discussion of the choice of the weights on a particular example

Finally, to illustrate the effect of η to the complexity of Perfect Simulation algorithm, we consider the age dependent Hawkes process with hard refractory period of length δ. We consider the setting of Proposition 4 and assume moreover that $\psi_i(.)$ is an L-Lipschitz function, where L is the Lipschitz constant. In this case we obtain an explicit upper bound for $\Delta_k^i(x)$, for any $x \in \mathcal{X}^{>\delta}$.

Proposition 9. For any $k \geq 1$, we have

$$\Delta_k^i(x) \leq L \times \left[\sum_{j\in V_i(k)\setminus V_i(k-1)} \sum_{0\leq m<k} h_{ji}(m\delta) + \sum_{j\in V_i(k-1)} h_{ji}((k-1)\delta)\right].$$

To prove this proposition we use the following lemma, which is a particular case of Lemma 2.4 in [20].

Lemma 1. For any $0 \le k < l$, $j \in \mathbf{I}$, $z \in \mathcal{X}^{>\delta}$, $t \in \mathbb{R}$ we have

$$\int_{t-l\delta}^{t-k\delta} h_{ji}(t-s)dz_s^j \le \sum_{k\le m<l} h_{ji}(m\delta).$$

Proof of Lemma 1. For any j, fixed $\epsilon > 0$, we have

$$\begin{aligned}\int_{[t-l\delta,t-k\delta-\epsilon]} h_{ji}(t-s)dz_s^j &= \sum_{k\le m<l} \int_{[t-(m+1)\delta-\epsilon,t-m\delta-\epsilon]} h_{ji}(t-s)dz_s^j \\ &\le \sum_{k\le m<l} h_{ji}(m\delta+\epsilon)\end{aligned}$$

by Assumption 3 and the fact that there is at most one jump in the interval of length δ. Therefore,

$$\begin{aligned}\int_{[t-l\delta,t-k\delta)} h_{ji}(t-s)dz_s^j &= \lim_{\epsilon\downarrow 0} \int_{[t-l\delta,t-k\delta-\epsilon]} h_{ji}(t-s)dz_s^j \\ &\le \lim_{\epsilon\downarrow 0} \sum_{k\le m<l} h_{ji}(m\delta+\epsilon) \\ &\le \sum_{k\le m<l} \lim_{\epsilon\downarrow 0} h_{ji}(m\delta+\epsilon) \\ &\le \sum_{k\le m<l} h_{ji}(m\delta),\end{aligned}$$

by using the theorem of monotone convergence and the fact that $h_{ji}(.)$ is a decreasing function according to Assumption 3. This completes the proof of Lemma 1. □

Proof of Proposition 9. By (ii) of Assumption 3, we have

$$\Delta_i^k(x) \le L \times \left[\sum_{j\in V_i(k)\setminus V_i(k-1)} \int_{-k\delta}^{0} h_{ji}(-s)dx_s^j + \sum_{j\in V_i(k-1)} \int_{-k\delta}^{-k\delta+\delta} h_{ji}(-s)dx_s^j \right].$$

Applying Lemma 1 we conclude that

$$\Delta_k^i(x) \le L \times \left[\sum_{j\in V_i(k)\setminus V_i(k-1)} \sum_{0\le m<k} h_{ji}(m\delta) + \sum_{j\in V_i(k-1)} h_{ji}((k-1)\delta) \right]$$

which ends the proof. □

Remark 17. In addition, from (i) of Assumption 3, we have

$$\sum_{j\in V_i(k)\setminus V_i(k-1)} \sum_{0\le m<k} h_{ji}(m\delta) + \sum_{j\in V_i(k-1)} h_{ji}((k-1)\delta) \to 0$$

as $k \to \infty$. Therefore $\Delta_k^i(x) \to 0$ when $k \to \infty$.

In what follows, to simplify the computation, we consider that $\mathbf{I} = \mathbb{Z}$, and for all i, we set

1. $h_{ji}(t) = \beta_{ji} \exp(-\alpha t)$ where β_{ji}, α are positive constants for all j, i. In addition, we take $\alpha = \frac{1}{\delta}$ and $\beta_{ji} = \frac{1}{2|j-i|^\gamma}$ for $j \neq i$ with a positive number γ and $\beta_{ii} = 1$.

2. $V_i(0) = \emptyset, V_i(1) = \{i\}, \dots, V_i(k) = \{i-k+1, \dots, i, i+1, \dots, i+k-1\}$ $\forall k \geq 2$.

3. $\eta_k^i = \eta_k = c_\eta \frac{1}{k^p}, \forall k \geq 1$, where p is a positive constant and c_η is a normalization constant.

From Proposition 9, we can choose

$$\begin{aligned}\Gamma_i &= \sup\left\{\frac{L}{c_\eta}, \sup_{k\geq 2}\left[\frac{Lk^p}{c_\eta}\left(\frac{1-e^{-k}}{1-e^{-1}}\sum_{j\in\{i-k+1, i+k-1\}}\beta_{ji} + e^{-k+1}\sum_{j\in V_i(k-1)}\beta_{ji}\right)\right]\right\} \\ &= \sup\left\{\frac{L}{c_\eta}, \sup_{k\geq 2}\left[\frac{Lk^p}{c_\eta}\left(\frac{1-e^{-k}}{(k-1)^\gamma(1-e^{-1})} + e^{-k+1}\left(1+\sum_{m=1}^{k-2}\frac{1}{m^\gamma}\right)\right)\right]\right\} := \Gamma.\end{aligned}$$

Since we define η_k^i are independent of i, so we can consider that all points have the same type. Instead of comparing the Backward Steps to a multitype branching process, we may therefore put $K(k) := K_i(k) := \sum_j K_i^j(k)$ and $W(n) := W_i(n) := \sum_j W_i^j(n)$ such that $K(k)$ is the total number of ancestors in the k^{th} set of ancestors and $W(n)$ is the total number of ancestors in the first n set of ancestors. Hence, all the vector $K(k) := K_i(k)$ and $W(n) = W_i(n)$ in Proposition 8 now are numbers.

Moreover, $v_k^i = V_i(k) \times [-k\delta, 0)$ and we set

$$\begin{aligned}\zeta &= \sum_{k'\geq 1} P(v_{k'}^i)\lambda(v_{k'}^i) = \sum_{k'\geq 1}\eta_{k'}\, k'\delta \sum_{j\in V_i(k')}\Gamma_j \\ &= \sum_{k'\geq 1} c_\eta \frac{1}{k'^p}\, k'\delta(2k'-1)\Gamma := f(p).\end{aligned}$$

By a classical well-known result in Branching processes [1] , we have for $k \geq 1$

$$\mathbb{E}(K(k)) = \zeta^k.$$

Therefore, the total expected number of simulated points $W := W(\infty)$ is given by

$$\mathbb{E}(W) = \frac{1}{1-\zeta}.$$

Now we are looking for values of p such that:

(a) $(\eta_k)_k$ defines a probability,

(b) the algorithm stays in $\mathscr{Y}$,

(c) the Kalikow decomposition exists (Proposition 4),

(d) the branching process goes extinct in finite time almost surely.

Once all these condition are fulfilled, we choose p such that it minimizes $f(p)$, so that the total number of simulated points is minimal.

$$\begin{aligned}
(a) &: \sum_k \eta_k = 1 \Rightarrow p > 1 \\
(b) &: \Gamma < \infty \Rightarrow p < \gamma \\
(c) &: \sum \int h_{ji} < \infty \Rightarrow \gamma > 1 \\
(d) &: \zeta < 1 \Rightarrow p - 2 > 1
\end{aligned}$$

Finally, we conclude that $3 < p < \gamma$. Moreover, we want to choose p such that $f(p)$ to be smallest possible, which means that we need to take p largest possible. Hence, we take p close to γ.

3.6 Conclusion

Prior research has investigated the Perfect Simulation based on a conditional Kalikow decomposition in continuous time point processes [14]. However, it is impossible to implement this study in practice. In the present study, we continue our work in [19], to extend the class of point processes which satisfies a new Kalikow decomposition. This decomposition plays a vital role to build a tractable algorithm in practice. In addition, we also improve the results in [14] and [19] on the existence of the Kalikow decomposition. Most notably, this is the first study to our knowledge to establish a Kalikow decomposition for a stochastic intensity in general context. Our results provide a general method to write a Kalikow decomposition for large class of point processes, for example: linear Hawkes processes, exponential Hawkes processes and including very complex Hawkes processes: age dependent Hawkes processes. However, some limitations are worth noting. Although, we succeed to write the Kalikow decomposition for a variety of Hawkes processes, we still have to restrict ourselves to a bounded intensity to implement the Perfect Simulation. This is due to the simulation of the first jump in the Backward Steps. Future work should focus on removing totally the upper bound of the intensity and extending the Perfect Simulation to the unbounded intensities.

Acknowledgements

This paper is dedicated to the Institute of Mathematics Hanoi, where the author was honored to work for more than three years. The author would like to express his very great appreciation to his supervisors Patricia Reynaud-Bouret and Eva Löcherbach for their patient guidance, enthusiastic encouragement and useful critiques on this research.

This work was supported by the French government, through the UCAJedi and 3IA Côte d'Azur Investissements d'Avenir managed by the National Research Agency (ANR-15- IDEX-01 and ANR- 19-P3IA-0002), by the CNRS through the "Mission pour les Initiatives Transverses et Interdisciplinaires" (Projet DYNAMO, "APP Modélisation du Vivant"), by the interdisciplinary Institute for Modeling in Neuroscience and Cognition (NeuroMod) of the Université Côte d'Azur, and directly by the National Research Agency (ANR-19-CE40-0024) with the ChaMaNe project.

References for Chapter 3

1. K. B. Athreya, P. E. Ney, and P. Ney. *Branching processes*. Courier Corporation, 2004.
2. E. Bacry, S. Delattre, M. Hoffmann, and J.-F. Muzy. "Some limit theorems for Hawkes processes and application to financial statistics". *Stochastic Processes and their Applications* 123:7, 2013, pp. 2475–2499.
3. P. Brémaud. *Point processes and queues*. Martingale dynamics, Springer Series in Statistics. Springer-Verlag, New York-Berlin, 1981, pp. xviii+354. ISBN: 0-387-90536-7.
4. P. Brémaud and L. Massoulié. "Stability of nonlinear Hawkes processes". *Ann. Probab.* 24:3, 1996, pp. 1563–1588. ISSN: 0091-1798. DOI: 10.1214/aop/1065725193. URL: https://doi.org/10.1214/aop/1065725193.
5. L. Carstensen, A. Sandelin, O. Winther, and N. R. Hansen. "Multivariate Hawkes process models of the occurrence of regulatory elements". *BMC bioinformatics* 11:1, 2010, pp. 1–19.
6. J. Chevallier. "Mean-field limit of generalized Hawkes processes". *Stochastic Processes and their Applications* 127:12, 2017, pp. 3870–3912.
7. D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes. Vol. I*. Second. Probability and its Applications (New York). Elementary theory and methods. Springer-Verlag, New York, 2003, pp. xxii+469. ISBN: 0-387-95541-0.
8. S. Delattre, N. Fournier, and M. Hoffmann. "Hawkes processes on large networks". *The Annals of Applied Probability* 26:1, 2016, pp. 216–261.
9. V. Didelez. "Graphical models for marked point processes based on local independence". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70:1, 2008, pp. 245–264.
10. A. Galves and E. Löcherbach. "Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets". *J. Stat. Phys.* 151:5, 2013, pp. 896–921. ISSN: 0022-4715. DOI: 10.1007/s10955-013-0733-9. URL: https://doi.org/10.1007/s10955-013-0733-9.
11. E. C. Hall and R. M. Willett. "Tracking dynamic point processes on networks". *IEEE Transactions on Information Theory* 62:7, 2016, pp. 4327–4346.
12. A. G. Hawkes. "Point spectra of some mutually exciting point processes". *J. Roy. Statist. Soc. Ser. B* 33, 1971, pp. 438–443. ISSN: 0035-9246. URL: http://links.jstor.org/sici?sici=0035-9246(1971)33:3%3C438:PSOSME%3E2.0.CO;2-G&origin=MSN.
13. A. G. Hawkes. "Spectra of some self-exciting and mutually exciting point processes". *Biometrika* 58, 1971, pp. 83–90. ISSN: 0006-3444. DOI: 10.1093/biomet/58.1.83. URL: https://doi.org/10.1093/biomet/58.1.83.
14. P. Hodara and E. Löcherbach. "Hawkes processes with variable length memory and an infinite number of components". *Adv. in Appl. Probab.* 49:1, 2017, pp. 84–107. ISSN: 0001-8678. DOI: 10.1017/apr.2016.80. URL: https://doi.org/10.1017/apr.2016.80.
15. S. Kalikow. "Random Markov processes and uniform martingales". *Israel J. Math.* 71:1, 1990, pp. 33–54. ISSN: 0021-2172. DOI: 10.1007/BF02807249. URL: https://doi.org/10.1007/BF02807249.

16. C. Mascart, A. Muzy, and P. Reynaud-Bouret. "Discrete event simulation of point processes: A computational complexity analysis on sparse graphs". *arXiv preprint arXiv:2001.01702*, 2020.

17. Y. Ogata. "On Lewis' simulation method for point processes". *IEEE Transactions on Information Theory* 27:1, 1981, pp. 23–31. DOI: 10.1109/TIT.1981.1056305.

18. G. Ost and P. Reynaud-Bouret. "Sparse space-time models: concentration inequalities and Lasso". *Ann. Inst. Henri Poincaré Probab. Stat.* 56:4, 2020, pp. 2377–2405. ISSN: 0246-0203. DOI: 10.1214/19-AIHP1042. URL: https://doi.org/10.1214/19-AIHP1042.

19. T. C. Phi, A. Muzy, and P. Reynaud-Bouret. "Event-scheduling algorithms with Kalikow decomposition for simulating potentially infinite neuronal networks". *SN Computer Science* 1:1, 2020, p. 35.

20. M. B. Raad, S. Ditlevsen, and E. Löcherbach. "Stability and mean-field limits of age dependent Hawkes processes". *Ann. Inst. Henri Poincaré Probab. Stat.* 56:3, 2020, pp. 1958–1990. ISSN: 0246-0203. DOI: 10.1214/19-AIHP1023. URL: https://doi.org/10.1214/19-AIHP1023.

21. P. Reynaud-Bouret and E. Roy. "Some non asymptotic tail estimates for Hawkes processes". *Bull. Belg. Math. Soc. Simon Stevin* 13:5, 2006, pp. 883–896. ISSN: 1370-1444. URL: http://projecteuclid.org/euclid.bbms/1170347811.

22. P. Reynaud-Bouret and S. Schbath. "Adaptive estimation for Hawkes processes; application to genome analysis". *Ann. Statist.* 38:5, 2010, pp. 2781–2822. ISSN: 0090-5364. DOI: 10.1214/10-AOS806. URL: https://doi.org/10.1214/10-AOS806.

23. G. Scarella, C. Mascart, A. Muzy, T. C. Phi, and P. Reynaud-Bouret. "Reconstruction de la connectivité fonctionnelle en Neurosciences: une amélioration des algorithmes actuels". In: *52èmes Journées de Statistique de la Société Française de Statistique (SFdS)*. 2021.

4 New methods for simulating point processes.

This chapter is written in collaboration with Paul Gresland during his master internship, under supervision of Patricia Reynaud-Bouret and Alexandre Muzy.

Status: working progress.

4.1 Introduction

Simulating neural networks has received a lot of attention in recent years, for example, there exist several grand projects such as the Human Brain Project in Europe, the Brain Mapping in Japan and the Brain Initiative in the United States. A complete simulated model will help reduce the cost of performing biological experiments and create an important premise before the actual experiments. In practice, there are many approaches to simulate neural networks, to cite but a few [1, 15, 23]. Here, we will focus on the point process approach, especially the simulation algorithm using the classical Ogata's thinning method [16]. However, it has been found to be too expensive for the huge network, see [14]. Instead, Mascart et al [14] considered a sparse network, that is a node is only connected to only few other nodes. This algorithm [14] can not apply to a complete graph with a huge number of nodes.

Another way to handle this problem is to combine the classical Ogata algorithm and Kalikow decomposition [13]. There has been extensive research regarding this combination for theoretical purposes. For example, in probability theory, for discrete-time processes, the authors of [4, 5, 6] developed a Perfect simulation algorithm by using Kalikow decomposition (see also Galves et al. [7, 8]). However, although the numerous studies in discrete time, little attention has been paid to the continuous-time processes. Up to our knowledge, there only exists one theoretical result, which was developed in [12]. Despite the excellent result in theory, their algorithm is not feasible in practice (see the discussion in [19] and also Chapter 2). Another notable limitation of [12] is the boundedness hypothesis of the intensities. Recently, Phi et al [19](see also Chapter 2 of this thesis) showed an interesting result by using a new type of Kalikow decomposition. More precisely, we have successfully simulated the activity of one neuron embedded in an infinite network without simulating the whole network. However, it rather focuses on simulating the activity of a part of neural networks than the whole network.

In this chapter, we will focus on simulating the neural networks with a large but not infinite number of neurons and for mathematical interest, we focus on the case where no deterministic upper bound for intensities exists at prior. This chapter is organized as follows. In the first section, we give the mathematical definitions and several useful notation. In Section 4.3, we present four algorithms, two are modifications of the classical Ogata's algorithm, two are the combination of Ogata's algorithm and Kalikow decomposition. All the proposed algorithms in this section are written for point processes having deterministically upper bounded intensity. In Section 4.4, we provide a modification of Ogata's algorithm and a combination of Ogata and Kalikow decomposition when such a deterministic bound does not exist. In Section 4.5, several numerical results are presented.

4.2 Mathematical definitions and notation

We consider N point processes in the time interval $[0, t_{max}]$ and assume that no events occurred before the time 0. Let $\mathbf{I}$ be the set of indices of these processes, $\mathbf{I} = \{1, 2, \dots, N\}$. Denote Z^i the counting process associated with index i. For any $0 \leq s < t < t_{max}$, $Z^i((s,t])$ records the number of points produced by the process Z^i during the interval $(s,t]$. Denote $\mathscr{F}$ a history of the process $Z = (Z^i)_{i \in I}$ (see Introduction of the thesis, Chapter 1). Let us recall that $(\phi_i(t|\mathscr{F}_t))_t$ is the intensity function of process Z^i.

4.2.1 Deterministic and stochastic upper bound of intensities

To the end of this chapter, we focus on two different assumptions of intensity. Either the intensity is bounded by a deterministic number, that is

Assumption 7 (Deterministically bounded intensity)**.** There exist deterministic positive numbers $(M_i)_i$ that dominate the intensities $(\phi_i(t|\mathcal{F}_t))_t$ respectively, namely

$$\phi_i(t|\mathcal{F}_t) \leq M_i \quad \forall i \in \mathbf{I}, \forall t \in [0, t_{max}].$$

For any arrival time T, let us denote $n_Z(T)$ the next arrival time after T of the process Z

$$n_Z(T) = \inf\{T' \in Z \, \text{such that} \, T' > T\}$$

and we also consider that

$$L_Z(t) = \sup\{T \in Z \, \text{such that} \, T < t\}$$

the last arrival time before time t of the process Z, with the convention that 0 is the last arrival time before the first event of any process Z.

The second assumption consists in saying that the intensity now is bounded by a particular stochastic predictable function. In this case, we construct a dominating intensity point after point, gradually starting from a initial point $T = 0$. I remove the measuring in the following assumption since we have not constructed the extended filtration. We have not constructed Π and don't have the notations X.

Assumption 8 (Stochastically bounded intensity)**.** There exists a dominating point process $\Pi = (\Pi^i)_i$ such that

1. At any T of the dominating process Π, there exists $M_i(T)$ a random variable such that

$$\sup_{t>T} \phi_i(t|\mathcal{F}_t)\mathbb{1}_{Z((T,t))=0} \leq M_i(T) < \infty \tag{4.2.1}$$

2. For each index i, Π^i has intensity

$$\Lambda_i(t) = \sum_T M_i(T)\mathbb{1}_{T<t\leq n_\Pi(T)}. \tag{4.2.2}$$

Note that, $\Lambda_i(t) \geq \phi_i(t|\mathcal{F}_t)$ almost surely and that therefore one might deduce points of Z by thinning points of Π. Intuitively, as soon as no point is produced, the intensity function is bounded by a (bounded) predictable function. That allows us to construct an upper bound to find a candidate for the next arrival time $n_\Pi(T)$, that is needed for the thinning algorithm, which will be introduced in the upcoming section.

4.2.2 Point structure

All the algorithms described below consist in generating points according to a dominating process Π^i, which will be

- either a Poisson process with intensity M_i if Assumption 7 is satisfied,
- or a more intricate dominating process Π^i of stochastic intensity Λ_i if Assumption 8 is satisfied.

In both cases, these points are going to be accepted or rejected according to various marks or uniform variables that are drawn at each step.

This is the reason why we define a *point* (written in italic) as a vector whose components are given in the following table. The number of the components can be extended if needed. For example, in the second part, we need to draw several uniform random variables. Also, for the choice of the neighborhood in Kalikow decomposition, it is more convenient to add V_T, the neighborhood picked at time T.

Component	Terminology	Meaning
$T \in [0, t_{max}]$	Time definition	When the *point* occurred
$j_T \in \mathbf{I}$	Index definition	Where the *point* occurred.
$X_T \in \{0, n.a, 1\}$	Thinning mark	indicates if the *point* is rejected ($X_T = 0$), selected ($X_T = 1$) or still unknown ($X_T = n.a$).
$U_T \in [0, 1]$	Uniform	Uniform random variable in [0, 1].
V_T	Neighborhood	Random neighborhood associated to the *point*, only needed if we use Kalikow decomposition.

Table 4.1: Definition of a *point*

A *point* S^i_T is of index i and at time T. Saying that a *point* S^i_T is produced means that the simulation of the Poisson process Π^i has an event at time T. As soon as S^i_T is produced, it is said to be unknown ($X_T = n.a$) and U_T is randomly set as an Uniform random variable on [0, 1].

A produced *point* S^i_T is said to be known if the thinning mark $X_T \neq n.a$. More precisely S^i_T is said to be accepted if $X_T = 1$ and rejected if $X_T = 0$. For instance, in the Ogata's algorithm for the deterministic bounded intensity, X_T is defined as the following

$$X_T = \begin{cases} 1 & \text{if} \quad U_T \leq \mathbb{P}(X_T = 1|\mathcal{F}_T), \\ 0 & \text{otherwise.} \end{cases} \tag{4.2.3}$$

where $\mathbb{P}(X_T = 1|\mathcal{F}_T)$ is the ratio between the intensity $\phi_i(T|\mathcal{F}_T)$ and M_i.
From now on, to simplify writing and to use notation coming from neural networks, we say *spike* for an accepted *point*.

4.2.3 Neuron structure

With the thinning method, several processes are taken into account to produce and accept a *point* S^i_T. The analogy between a biological neuron and a point process reaches its limits. That is why we introduce a formal definition of a *neuron* (written in italic). A *neuron* is a mathematical structure that includes all the variables required to produce and accept the *points* as well as the produced *points* themselves (whether they are spikes that are accepted or not).

If a *point* S_T^i is produced, we also say that the *neuron* i produced this *point*. In the same way if *point* S_T^i is accepted, we also say that the *neuron* i accepted this *point* or equivalently that the *neuron* i produced the *spike* S_T^i.

The component of a *neuron* i is given in the following table. These following components are designed for the bounded intensity case.

Component	Meaning
$i \in \mathbf{I}$	Index of the *neuron*.
$M_i \in \mathbb{R}_+$	Upper bound of the firing rate of the *neuron* (Assumption 7)
$n_i \in \mathbb{N}$	Number of *points* produced by the *neuron*, which obeys a Poisson distribution with parameter $M_i\, t_{max}$.
$(T_k^i)_{k\in[\|1,n_i\|]} \in \mathbb{R}^{n_i}$	Vector of times of the *points* produced by the *neuron*
$\mathcal{S}^i = (S_T^i)_{T\in(T_k^i)_k}$	Vector of the n_i *points* produced by the *neuron*
$(U_T)_{T\in(T_k^i)} \in [0,1]^{n_i}$	Vector of the uniform random variables each linked to a unique *point* produced by the *neuron*
$(X_T)_{T\in(T_k^i)} \in \{0, n.a, 1\}^{n_i}$	Vector of the thinning marks of the *points* produced by the *neuron*

Table 4.2: Definition of a *neuron*

4.3 First part: Deterministically bounded intensities and Parallelization

This section is devoted to present four algorithms: two based on Ogata's thinning methods and the two other methods being a combination of Ogata's algorithm and Kalikow decomposition. To simplify, we write our algorithms in a particular example: the Linear Hawkes processes with refractory period [3, 20]. The general algorithms can be constructed very similarly, we give more details in the remarks at the end of each algorithm (if needed). Recall that the intensity of Linear Hawkes process with hard refractory period is given by

$$\phi_i(t|\mathcal{F}_t) = \left(\mu_i + \sum_{j\in\mathbf{I}} \int_0^t h_{ji}(t-s)dZ_s^j\right)\mathbb{1}_{t-L_{Z^i}(t)>\delta} \tag{4.3.1}$$

where recall that $L_{Z^i}(t) = sup\{T \in Z^i \text{ such that } T < t\}$, μ_i's positive real numbers and h_{ji} the non negative functions with support in $\mathbb{R}_+$.

We begin with two modifications of Ogata's algorithms. The first one called OtS, is the classical Ogata's thinning algorithm in [16]. The second, called OtpP, is a partially parallelized version of OtS.

4.3.1 Sequential Ogata's thinning algorithm

Ogata's thinning algorithm

The modificated Ogata's thinning method aims to simulate all the processes $i \in \mathbf{I}$ in two steps: seeding step and thinning step:

i. Seeding step : For each index $i \in \mathbf{I}$, a list of candidate points is built by simulating Π^i a homogeneous Poisson process with intensity M_i defined in Assumption 7.

ii. Thinning step: Each of the candidate points are selected or rejected via an exact computation of the intensity at the corresponding time.

The algorithm presented in this section is a rewriting of Ogata's algorithm [16] in accordance with our programming method.

Remark 18. This method is indeed a modification of Ogata's algorithm [16] since in the original paper, Ogata constructs a dominating intensity process point after point before performing the thinning step. Throughout this chapter, whenever we say Ogata's algorithm, it refers to this modification of Ogata's algorithm, and not the original one in [16].

Overall description

As this algorithm is essentially the Ogata thinning method (Section 4.3.1), it is also written in two main steps.

During the first step each *neuron* $i \in \mathbf{I}$ produced the set of *points* $\mathcal{S}^i$. In particular, the vector $(T^i_k)_{k\in[|1,n_i|]}$ corresponds to the times arrival of a homogeneous Poisson process with intensity M_i. Then the whole set of *points* $\mathcal{S}$ is built as the unions of all sets $\mathcal{S}^i$. The set $\mathcal{S}$ is sorted to finally contain all the *points* in an increasing T-order, that is S^i_T is "smaller" than $S^j_{T'}$ if and only if $T < T'$.

During the second step, for each *point* S^i_T in the set $\mathcal{S}$, the intensity $\phi_i(T|\mathcal{F}_T)$ is computed using the Hawkes formula with refractory period (4.3.26). The algorithm splits this formula to isolate the Hawkes process intensity $\psi_i(T|\mathcal{F}_T)$ from the refractory period constraint:

$$\phi_i(T|\mathcal{F}_T) = \psi_i(T|\mathcal{F}_T)\mathbb{1}_{T-L_{Z^i}(T)>\delta} \tag{4.3.2}$$

where

$$\psi_i(T|\mathcal{F}_T) = \psi_i(T|\mathcal{S}_T) = \mu_i + \sum_{j\in\mathbf{I}} \sum_{T'\in Z^j} h_{ji}(T-T')\mathbb{1}_{X_{T'}=1}. \tag{4.3.3}$$

If $\mathbb{1}_{T-L_{Z^i}(T)>\delta} = 0$, the refractory period is not respected, $\phi_i(T|\mathcal{F}_T) = 0$ without the need to compute $\psi_i(T|\mathcal{F}_T)$. On the other hand, if $\mathbb{1}_{T-L_{Z^i}(T)>\delta} = 1$ then $\phi_i(T|\mathcal{F}_T) = \psi_i(T|\mathcal{F}_T)$ only depends on $\mathcal{S}_T$, the set of the *points* produced before T. Since we treat each point in increasing T-order, we know if the T' in 4.3.3 satisfy $X_{T'} = 1$ or not.

Remark 19. We define a relation " $\preceq$ " between two configurations of thinning marks on $\mathcal{S}_T$: $\mathcal{S}_T^1$ and $\mathcal{S}_T^2$. We say $\mathcal{S}_T^1 \preceq \mathcal{S}_T^2$ if all the accepted point in $\mathcal{S}_T^1$ is also included and accepted in $\mathcal{S}_T^2$, that is

$$\forall S_T^i \in \mathcal{S}_T^1, \text{ if } X_T = 1 \text{ then } S_T^i \in \mathcal{S}_T^2$$

It is worth noting that $\psi_i(T|\mathcal{S}_T)$ is increasing with respect to the configuration of thinning marks on $\mathcal{S}_T$, that is

$$\psi(T|\mathcal{S}_T) \leq \psi(T|\mathcal{S}_T')$$

whenever $\mathcal{S}_T \preceq \mathcal{S}_T'$. It also implies that the intensity function $\phi_i(T|\mathcal{S}_T)$ is increasing with respect to the configuration of thinning marks on $\mathcal{S}_T$.

Knowing $\phi_i(T|\mathcal{F}_T)$ allows computing the probability to select the *point* S_T^i using the thinning formula

$$\mathbb{P}(X_T = 1|\mathcal{F}_T) = \frac{\phi_i(T|\mathcal{F}_T)}{M_i}. \tag{4.3.4}$$

Then the thinning mark X_T can be defined as a realization of a random variable Bernoulli with parameter $\mathbb{P}(X_T = 1|\mathcal{F}_T)$ (formula 4.2.3).

Once every *point* S_T^i gets its final thinning mark, the *spikes* train is obtained by keeping only the selected points.

Pseudo-code

Algorithm 4 Algorithm OtS (for Ogata's thinning Sequential)

▷ **Step 1: Seeding step**
for each *neuron i* **do**
  Compute the number of *points* : n_i follows a $Poisson(M_i\, t_{max})$.
  for each *point* indexed by $k \leq n_i$ **do**
    Simulate the time : T^i_k follows a $U([0, t_{max}])$ distribution
    Define the uniform : $U_{T^i_k}$ follows a $U([0,1])$ distribution
    Set the *point* as unknown : the thinning mark $X_{T^i_k} = n.a$
  end for
end for
▷ **Step 2: Thinning**
Define the whole set of *points* : $\mathscr{S} = \bigcup \mathscr{S}^i$
Sort this set of *points* $\mathscr{S}$ in an increasing T-order
for each point $S^i_T \in \mathscr{S}$ **do**
  Compute the intensity $\phi_i(T|\mathscr{F}_T)$ with formula (4.3.2)
  Compute the probability $\mathbb{P}(X_T = 1|\mathscr{F}_T) = \dfrac{\phi_i(T|\mathscr{F}_T)}{M_i}$ to accept S^i_T
  Determine the thinning mark X_T of the *point* S^i_T (4.2.3)
end for
▷ **Final step : format the spikes trains**
Delete the points that have a thinning mark $X_T = 0$

Remark 20. The algorithm OtS stops after a finite number of steps almost surely. Indeed only a finite number of *points* are produced after the seeding step.

4.3.2 Partially parallelized Ogata's thinning algorithm

Remark 21. The algorithm OtS cannot be parallelized without an adaptation. Indeed, at line 12 of Algorithm 4 to compute $\phi_i(T|\mathcal{F}_T)$, we need the past before time T, $\mathcal{F}_T$. This past is available only if the *points* in $\mathcal{S}$ are sorted in time and we go through point by point. In the case of parallel computation, the *points* are at best partially sorted. Hence when a *point* S_T is computed it may exist $S_{T'}$ with $T' < T$ where $X_{T'} = n.a.$

To deal with the problem in Remark 21, let us note that even if the intensity $\phi_i(T|\mathcal{F}_T)$ cannot be determined, it remains possible to provide nontrivial bounds for $\psi_i(T|\mathcal{S}_T)$, the part without the refractory period. Indeed, we already note (Remark 19) that $\psi_i(T|\mathcal{S}_T)$ is increasing with respect to any thinning mark of the *points* produced before T. Hence it should be possible to build two sets $\mathcal{S}_T^{min}$ and $\mathcal{S}_T^{max}$ of *points* all known (that is none with the thinning mark n.a) such that

$$\psi_i(T|\mathcal{S}_T^{min}) \leq \psi_i(T|\mathcal{S}_T) \leq \psi_i(T|\mathcal{S}_T^{max}). \tag{4.3.5}$$

For any *point* $S_{T'} \in \mathcal{S}_T$, we define the *points* $S_{T'}^{min}$ and $S_{T'}^{max}$ as the *point* $S_{T'}$ with the respective thinning mark $X_{T'}^{min}$ and $X_{T'}^{max}$ such that

$$X_{T'}^{min} = \begin{cases} 0 \text{ if } X_{T'} = n.a \\ X_{T'} \text{ otherwise} \end{cases} \tag{4.3.6}$$

$$X_{T'}^{max} = \begin{cases} 1 \text{ if } X_{T'} = n.a \\ X_{T'} \text{ otherwise} \end{cases} \tag{4.3.7}$$

The sets $\mathcal{S}_T^{min} = \bigcup S_{T'}^{min}$ and $\mathcal{S}_T^{max} = \bigcup S_{T'}^{max}$ satisfy the relation (4.3.5). Our goal is to use the bounds for $\psi_i(T|\mathcal{S}_T)$ to determine bounds for the acceptance probability of the *point* S_T^i. We are looking for $\mathbb{P}_{min}(X_T = 1|\mathcal{F}_T)$ and $\mathbb{P}_{max}(X_T = 1|\mathcal{F}_T)$ such that

$$\mathbb{P}_{min}(X_T = 1|\mathcal{F}_T) \leq \mathbb{P}(X_T = 1|\mathcal{F}_T) \leq \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T). \tag{4.3.8}$$

Since the relation (4.3.5) provides bounds for $\psi_i(T|\mathcal{F}_T)$ and not for $\phi_i(T|\mathcal{F}_T)$ we have to discuss three possible cases concerning the refractory period.

i. Case 1 : The refractory period is not respected if $\exists S_{T'}^i$ with $T - \delta \leq T' < T$ and $X_{T'} = 1$. In this case S_T^i is surely rejected ($X_T = 0$):

$$\mathbb{P}_{min}(X_T = 1|\mathcal{F}_T) = \mathbb{P}(X_T = 1|\mathcal{F}_T) = \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T) = 0. \tag{4.3.9}$$

ii. Case 2 : The respect of refractory period cannot be determined if $\exists S_{T'}^i$ with $T - \delta \leq T' < T$ and $X_{T'} = n.a.$ In this case :

$$\begin{cases} \mathbb{P}_{min}(X_T = 1|\mathcal{F}_T) = 0 \\ \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T) = \frac{\psi_i(T|\mathcal{S}_T^{max})}{M_i}. \end{cases} \tag{4.3.10}$$

ii. Case 3 : The refractory period is respected if $\forall S^i_{T'}$, with $T-\delta \leq T' < T$ the thinning mark $X_{T'} = 0$. Hence $\phi_i(T|\mathcal{F}_T) = \psi_i(T|\mathcal{S}_T)$ then

$$\begin{cases} \mathbb{P}_{min}(X_T = 1|\mathcal{F}_T) = \frac{\psi_i(T|\mathcal{S}_T^{min})}{M_i} \\ \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T) = \frac{\psi_i(T|\mathcal{S}_T^{max})}{M_i}. \end{cases} \tag{4.3.11}$$

If the range of probability $[\mathbb{P}_{min}(X_T = 1|\mathcal{F}_T), \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T)]$ is small enough, this allows us to determine the thinning mark.

$$X_T = \begin{cases} 1 & \text{if} \quad U_T \leq \mathbb{P}_{min}(X_T = 1|\mathcal{F}_T) \\ 0 & \text{if} \quad U_T \geq \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T) \\ n.a & \text{otherwise.} \end{cases} \tag{4.3.12}$$

Overall description

To build a parallelized algorithm we introduce the structure *population* consisting on a subset of the set of the *neurons*. Let $\mathbf{K}$ be the set of index of N_K *populations*, $\mathbf{K} = \{1, \ldots, N_K\}$. The set of *populations* forms a partition of the set of *neurons*. This means that each *neuron* belongs to exactly one *population*. The algorithm computes sequentially each *neuron* within the same population and the *neurons* of two different *populations* are computed in parallel.

During the seeding step, the computation of each *population* is completely independent. Indeed this step consists in simulating independent Poisson processes for each *neuron*. $\mathcal{S}^k_{pop}$, the set of produced *points* for each *population* k is sorted in an increasing T-order. Then to prepare for the next step, the complete set of produced *points* is established as $\mathcal{S} = \bigcup_1^{N_K} \mathcal{S}^k_{pop}$.

The thinning step is an alternation between parallel computation and sequential information sharing. During a parallel phase, each *population* k goes through the set of *points* $\mathcal{S}^k_{pop}$ trying to determine as many thinning marks as possible. Then, the *populations* synchronize, each one waiting for the others to finish their computation. During the sequential phase, the whole set of *points* $\mathcal{S}$ is updated with the thinning marks determined by each *population*. The alternation between the parallel and sequential phase is operated as long as there are still unknown *points*.

Pseudo-code

Algorithm 5 Algorithm OtpP (for Ogata's thinning partially parallelized)

▷ **Step 1: Seeding step**
1: **for** each *population* k **in parallel do**
2:     **for** each *neuron* $i \in$ population **do**
3:         Compute the number of *points* : n_i follows a $Poisson(M_i\, t_{max})$.
4:         **for** each *point* indexed by $l \leq n_i$ **do**
5:             Simulate the time : T_l^i follows a $U([0, t_{max}])$ distribution
6:             Define the uniform : $U_{T_l^i}$ follows a $U([0,1])$ distribution
7:             Set the *point* as unknown : the thinning mark $X_{T_l^i} = n.a$
8:         **end for**
9:     **end for**
10:     Define $\mathcal{S}_{pop}^k = \bigcup \mathcal{S}^i$ the whole set of *points* belonging to the *population*
11:     Sort this set of *points* $\mathcal{S}_{pop}^k = \bigcup \mathcal{S}^i$ in an increasing T-order
12: **end for**
13: Define the whole set of *points* : $\mathcal{S} = \bigcup \mathcal{S}_{pop}^k$
▷ **Step 2: Thinning**
14: **while** At least one *point* $S_T^i \in \mathcal{S}$ is unknown **do**
15:     **for** each *population* k **in parallel do**
16:         **for** each unknown *point* $S_T^i \in \mathcal{S}_{pop}^k$ **do**
17:             Compute the upper bound probability $\mathbb{P}_{max}(X_T = 1|\mathcal{F}_T)$ (4.3.8)
18:             Compute the lower bound probability $\mathbb{P}_{min}(X_T = 1|\mathcal{F}_T)$ (4.3.8)
19:             Determine the thinning mark X_T if it is possible (4.3.12)
20:         **end for**
21:     **end for**
22:     **for** each *population* k **in sequence do**
23:         Update the whole set of *points* $\mathcal{S}$ with the thinning marks
24:     **end for**
25: **end while**
▷ **Final step : format the spikes trains**
26: Delete the points that have a thinning mark $X_T = 0$

Remark 22. The algorithm OtpP stops after a finite number of steps almost surely. Indeed only a finite number of *points* are produced after the seeding step. Then at each WHILE loop, at least one *population* accepts or rejects a *point*. Indeed the *point* with the smallest time can always be accepted or rejected.

Remark 23. The algorithm OtpP is more general than the algorithm OtS (the sequential version). Indeed the algorithm OtpP with a single *population* is completely equivalent to the Algorithm OtS.

Remark 24. The algorithm OtpP is only partially parallelized. In particular, there is an alternation between a parallel loop (line 15) and a sequential loop (line 22). There is also a sequential computation (line 13) that separates the two steps of the thinning.

Remark 25. A crucial hypothesis that is needed for the algorithm OtpP works is the construction of $\mathbb{P}_{min}, \mathbb{P}_{max}$. In general, the algorithm OtpP works as soon the intensity is increasing with respect to the configuration of thinning marks, that is

$$\phi(T|\mathcal{S}_T) \leq \phi(T|\mathcal{S}'_T)$$

whenever $\mathcal{S}_T \preceq \mathcal{S}'_T$. In words, more accepted points more chances to emit a new spike.

The synchronization between the two steps seems to us unavoidable. On the other hand, it might be possible to avoid sequential computations within the second step. However, we have encountered difficulties with the use of the python *multiprocessing* package. When dynamic information was shared by parallel procedures, it generated synchronizations that we could not control. This is why, in the algorithm OtpP, the information managed by a $population$ is always inaccessible to the information managed by another. This of course has a cost, as we regularly interrupt parallel computation to consolidate and share information (line 22). Further discussion can be found in Section 4.5 of this chapter.

Remark 26. A criticism that can be made about the algorithm partially parallelizing Ogata's thinning is the presence of the WHILE loop (line 14). As far as we know it is not possible to get rid of it.

However, it remains possible to reduce the number of iterations using Kalikow decomposition. Since the intensity using Kalikow decomposition depends on a smaller number of past event, the probability range $[\mathbb{P}_{min}(X_T = 1|\mathcal{F}_T), \mathbb{P}_{max}(X_T = 1|\mathcal{F}_T)]$ to accept $point$ (4.3.8) should be smaller in many cases.

In addition, the advantages are not limited to the case of parallel computing. As mentioned earlier (Remark 27), Kalikow decomposition could allow making linear the execution time with respect to the number of neurons, even in the case of sequential computation. This is why the first algorithm we present is sequential. We called it KOtS for Kalikow-Ogata's thinning Sequential. In a second sub-section, we present a partially parallelized version called KOtpP for Kalikow-Ogata's thinning partially parallelized.

These two algorithms are also based on the Ogata thinning method, which is a building of the $spike$ trains in two steps. The first step produces the $points$ by simulation of Poisson processes (seeding step). The second step selects some of the $points$ to build the $spike$ trains (thinning step).

4.3.3 Sequential Kalikow-Ogata's thinning algorithm

Kalikow decomposition

In this section, by adding Kalikow decomposition to the classical Ogata algorithm, we introduce two new algorithms in order to reduce the complexity of algorithms in previous sections. The Kalikow decomposition used here is unconditional see [18] and also Chapter 2 and Chapter 3.
Recall that $\mathbf{V}_T$ a family of the neighborhood is a set of all possible neighborhoods v_T (we refer to the formal definition of the neighborhood in Chapter 2 and Chapter 3). V_T is a random variable, taking values in $\mathbf{V}_T$. We also assume that $\emptyset$ belongs to $\mathbf{V}_T$ for all T.

On the neighborhood family $\mathbf{V}_T$, we define a probability distribution $\lambda(.)$ such that $\mathbb{P}(V_T = v_T) = \lambda(v_T)$. For any neuron i, suppose that the intensity $\phi_i(t|\mathcal{F}_t)$ obeys a Kalikow decomposition for all $t \in [0, t_{max}]$. This means that, at each time t, there exists a neighborhood family $\mathbf{V}_t$, in which, we define a probability distribution $\lambda_i(v_t)$ and functions $\phi_i^{v_t}(t|\mathcal{F}_t)$ that depend only on the spikes in neighborhood v_t such that

$$\phi_i(t|\mathcal{F}_t) = \sum_{v_t \in \mathbf{V}_t} \lambda_i(v_t)\phi_i^{v_t}(t|\mathcal{F}_t).$$

To illustrate our result, we focus on a very specific choice of neighborhood v_T [1], which is given by:

$$v_T = \{j\} \times [T-(n+1)\delta, T-n\delta) \tag{4.3.13}$$

with $j \in I \bigcup \{\emptyset\}$ and $n \in \mathbb{N}$. With this concrete definition, the associated intensity $\phi_i^{v_T}(T|\mathcal{F}_T)$ is as follows

$$\phi_i^{v_T}(T|\mathcal{F}_T) = \begin{cases} \frac{\mu_i \mathbb{1}_{T-L_{Z^i}(T)>\delta}}{\lambda_i(v_T)} \text{ if } j = \emptyset \\ \frac{\mathbb{1}_{T-L_{Z^i}(T)>\delta}}{\lambda_i(v_T)} \int_{T-(n+1)\delta}^{T-n\delta} h_{ji}(T-t)dZ_t^j \text{ if } j \neq \emptyset. \end{cases} \tag{4.3.14}$$

Remark 27. The intensity $\phi_i^{v_T}(T|\mathcal{F}_T)$ (after picking the neighborhood) is simpler than $\phi_i(T|\mathcal{F}_T)$. It depends at most on only one pre-synaptic *neuron* and on a very small range of the past time. This is the most important motivation for using the Kalikow decomposition. It can be used to upper bound the number of past events that must be known to determine the selection or rejection of a *point*. If this upper bound remains constant whatever the number of neurons is, this could allow making linear the execution time with respect to the number of neurons. Moreover, as the Kalikow decomposition allows to simplify the dependencies, it is an efficient tool to implement a parallelized algorithm.

Remark 28. M_i, the upper bound of the intensity $\phi_i(T|\mathcal{F}_T)$ is not an upper bound of the intensity $\phi_i^{V_T}(T|\mathcal{F}_T)$. However, as mentioned in Chapter 2 in the case of the Hawkes process with refractory period, it remains possible to provide an upper bound of $\phi_i^{V_T}(T|\mathcal{F}_T)$.

The upper bound of $\phi_i^{V_T}(T|\mathcal{F}_T)$ depends on the interaction functions and the chosen decomposition. For the simulation purpose, let us assume that the upper bound exist.

Assumption 9. i. For any i, T and neighborhood $v_T \in \mathbf{V}_T$, there exists a deterministic positive number M_{v_T} such that

$$\phi_i^{v_T}(T|\mathcal{F}_T) \leq M_{v_T}.$$

ii. Grant the first hypothesis (i.), there exists moreover a deterministic positive number M_i such that

$$\sup_T \sup_{v_T \in \mathbf{V}_T} M_{v_T} \leq M_i.$$

Remark 29. Here for each neighborhood v_T, we suppose that there exists a deterministic upper bound M_{v_T} for $\phi_i^{v_T}(T|\mathcal{F}_T)$. This hypothesis eventually guarantees to do thinning on each neighborhood v_T.

[1]Note that, we write this neighborhood notation here informally. In fact, v_T in the previous chapter equal to $v_T \cup \{i\} \times (T-\delta, T)$

Finally, by assume the second hypothesis (ii.), it ensures that the thinning procedure is applicable for every neighborhood v_T.

Remark 30. In our case, a neighborhood is written as the Cartesian product of two subsets (4.3.13), one for space and the other for a time.

Thus, the probability $\lambda(v_T)$ can be written as the product of two probabilities, one in space, one in time:

$$\lambda(v_T) = \lambda^{space}(\{j\}) \times \lambda^{time}(\{n\}) \tag{4.3.15}$$

Remark 31. Note that, in our settings in Section 4.3.5, $\lambda^{space}(\{j\})$ and $\lambda^{time}(\{n\})$ can be computed independently of T.

Overall description

As this algorithm is based the Ogata thinning method (Section 4.3.1), it is also written in two main steps.

During the first step, as for the algorithm OtS, each *neuron* $i \in \mathbf{I}$ produced the set of *points* $\mathcal{S}^i$. In particular , the vector $(T_k^i)_{k\in[|1,n_i|]}$ corresponds to the time arrivals of a homogeneous Poisson process with intensity M_i.

In addition, a neighborhood v_T is randomly drawn for each *point* S_T^i. As notice in Remark 30, v_T is the Cartesian product of two independent subset of the past information. We decompose the neighborhood in a neighborhood in space $\{j\}$ and a neighborhood in time $[T-(n+1)\delta, T-n\delta]$ that are drawn independently, respectively with probability $\lambda^{space}(\{j\})$ and $\lambda^{time}(\{n\})$. Then, the knowedge of the neighborhood v_T allows to perform a first thinning, called a pre-thinning. Indeed, as mentioned in Remark 29, the intensity $\phi_i^{v_T}(T|\mathcal{F}_T)$ is upper bounded by a deterministic constant M_{v_T} smaller than M_i. This allows to eventually reject the *point* S_T^i by setting its thinning mark as follows.

$$X_T = \begin{cases} 0 & \text{if} \quad U_T > \frac{M_{v_T}}{M_i} \\ n.a & \text{otherwise.} \end{cases} \tag{4.3.16}$$

In the case of our implementation M_{v_T} only depends on the neighborhood in space $\{j\}$. For this reason, the algorithm checks first the *points* that are rejected then picks the neighborhoods in time only for still unknown *points*.

Then the whole set of *points* $\mathcal{S}$ is built as the unions of all sets $\mathcal{S}^i$. The set $\mathcal{S}$ is sorted to finally contain all the *points* in an increasing T-order.

During the second step, for each unknown *point* S_T^i in the set $\mathcal{S}$, the intensity $\phi_i^{v_T}(T|\mathcal{F}_T)$ is computed using the formula defined in the section about Kalikow decomposition (4.3.14). The algorithm split this formula to isolate the Hawkes process intensity $\psi_i^{v_T}(T|\mathcal{F}_T)$ from the refractory period constraint:

$$\phi_i^{v_T}(T|\mathcal{F}_T) = \psi_i^{v_T}(T|\mathcal{F}_T)\mathbb{1}_{T-L_{Z^i(T)}>\delta} \tag{4.3.17}$$

with $\psi_i^{v_T}$ defined as follows.

$$\psi_i^{v_T}(T|\mathcal{F}_T) = \begin{cases} \psi_i^{\emptyset}(T|\mathcal{F}_T) = \frac{\mu_i}{\lambda_i(v_T)} \text{ if } j = \emptyset \\ \frac{1}{\lambda_i(v_T)} \int_{T-(n+1)\delta}^{T-n\delta} h_{ji}(T-t) dZ_t^j \text{ if } j \neq \emptyset. \end{cases} \tag{4.3.18}$$

If $\mathbb{1}_{T-L_{Z^i}(T)>\delta} = 0$, the refractory period is not respected, $\phi_i^{v_T}(T|\mathcal{F}_T) = 0$ without the need to compute $\psi_i^{v_T}(T|\mathcal{F}_T)$. In the opposite case, if $\mathbb{1}_{T-L_{Z^i}(T)>\delta} = 1$ then $\phi_i^{v_T}(T|\mathcal{F}_T) = \psi_i^{v_T}(T|\mathcal{F}_T)$ only depends on $\mathcal{S}^{v_T}$, the set of the *points* produced inside the neighborhood v_T.

$$\psi_i^{v_T}(T|\mathcal{F}_T) = \psi_i^{v_T}(T|\mathcal{S}^{v_T}) = \begin{cases} \frac{\mu_i}{\lambda_i(\emptyset)} \text{ if } j = \emptyset \\ \frac{1}{\lambda_i(v_T)} \sum_{S_{T'}^j \in \mathcal{S}^{v_T}} h_{ji}(T-T') \mathbb{1}_{X_{T'}=1} \text{ if } j \neq \emptyset. \end{cases} \tag{4.3.19}$$

Remark 32. Again, similarly to Remark 19, we observe that $\psi_i^{v_T}(T|\mathcal{S}^{v_T})$ is increasing with respect to the configuration of thinning marks in v_T.

Remark 33. In the case where $j \neq \emptyset$, since the time neighborhood $[T-(n+1)\delta, T-n\delta]$ has a length equals to a refractory period δ, only one term of the sum is a non zero term. In other words, $\psi_i^{v_T}(T|\mathcal{S}^{v_T})$ depends on at most one previous event. It was mentioned within Remark 27 as a condition to make the execution time linear with respect to the number of *neurons*.

The algorithm can then compute the probability to select the *point* S_T^i using the thinning formula with the intensity $\phi_i^{v_T}(T|\mathcal{F}_T)$:

$$\mathbb{P}(X_T = 1 | V_T = v_T, \mathcal{F}_T) = \frac{\phi_i^{v_T}(T|\mathcal{F}_T)}{M_i}. \tag{4.3.20}$$

Then the thinning mark X_T can be defined as a realization of a random variable Bernoulli with parameter $\mathbb{P}(X_T = 1 | V_T = v_T, \mathcal{F}_T)$.

Once every *point* S_T^i get its final thinning mark, the *spikes* train is obtained by keeping only the selected *points*.

PSEUDO-CODE

Algorithm 6 Algorithm KOtS (for Kalikow-Ogata's thinning Sequential)

▷ **Step 1: Seeding step**
for each *neuron* i **do**
  Compute the number of *points* : n_i follows a $Poisson(M_i t_{max})$.
  for each *point* indexed by $k \leq n_i$ **do**
    Simulate the time : T_k^i follows a $U([0, t_{max}])$ distribution
    Define the uniform : $U_{T_k^i}$ follows a $U([0, 1])$ distribution
▷ **Step 2: Kalikow step and pre-thinning**
    Pick neighborhood in space : $\{j\}$ wp $\lambda_i^{space}(\{j\})$
    Compute the upper bound $M_{v_{T_k^i}}$ which only depends on j
    if $U_{T_k^i} > \frac{M_{v_{T_k^i}}}{M_i}$ **then**
      Reject the *point* : the thinning mark $X_{T_k^i} = 0$
    else
      Set the *point* as unknown : the thinning mark $X_{T_k^i} = n.a$
      Pick neighborhood in time : $[T_k^i - (n+1)\delta, T_k^i - n\delta]$ wp $\lambda_i^{time}(\{n\})$
    end if
  end for
end for
▷ **Step 3 : Thinning**
Define the whole set of *points* : $\mathcal{S} = \bigcup \mathcal{S}^i$
Sort this set of *points* $\mathcal{S}$ in an increasing T-order
for each point $S_T^i \in \mathcal{S}$ **do**
  Compute the intensity $\phi_i^{v_T}(T|\mathcal{F}_T)$ with formula (4.3.17)
  Compute the probability $\mathbb{P}(X_T = 1 | V_T = v_T, \mathcal{F}_T) = \frac{\phi_i^{v_T}(T|\mathcal{F}_T)}{M_i}$ to accept S_T^i
  Determine the thinning mark X_T of the *point* S_T^i (4.2.3)
end for
▷ **Final step : format the spikes trains**
Delete the points that have a thinning mark $X_T = 0$

Remark 34. For a general choice of neighborhood, it suffices to replace Step 2 by choosing a neighborhood $v_{T_k^i}$ with a general Kalikow decomposition and do pre-thinning for each T_k^i.

4.3.4 Partially parallelized Kalikow-Ogata's thinning algorithm

Remark 35. The problem is very similar to the case of algorithms without Kalikow decomposition.

We follow the same pattern as in Section 4.3.2 in which we proposed a parallelized version of Ogata's thinning algorithm. The main issue was to define bounds for the intensity if it cannot be exactly computed. Let us start with the intensity without considering the refractory period. We are looking for bounds for $\psi_i^{v_T}(T|\mathcal{S}^{v_T})$ if there exist a *point* $S_{T'}^j \in \mathcal{S}^{v_T}$ with the thinning mark $X_{T'} = n.a.$

By Remark 33 we know that a unique *point* of the neighborhood will be selected at the end. This remark simplifies the procedure. First, if a *point* $S^j_{T'}$ has already been selected we know that it is the only one and that the other *points* will be rejected. Hence $\psi_i^{\upsilon_T}(T|\mathcal{S}^{\upsilon_T})$ can be exactly computed as follows

$$\psi_i^{\upsilon_T}(T|\mathcal{S}_T) = \frac{h_{ji}(T-T')}{\lambda_i(\upsilon_T)}. \tag{4.3.21}$$

Otherwise there is no accepted *point* in $\mathcal{S}^{\upsilon_T}$, in this case, the lower bound of $\psi_i^{\upsilon_T}(T|\mathcal{S}_T)$ is 0. It remains to find the upper bound. We define a set $\mathcal{S}_T^{\upsilon_T,max}$ such that

$$0 \leq \psi_i^{\upsilon_T}(T|\mathcal{S}_T) \leq \psi_i^{\upsilon_T}(T|\mathcal{S}_T^{\upsilon_T,max}). \tag{4.3.22}$$

With the same remark as before we can build $\mathcal{S}_T^{\upsilon_T,max}$ as a set of a unique *point*, the one not rejected that maximize the interaction function h_{ji}, with j fixed by υ_T. Hence the bounds are as follows

$$0 \leq \psi_i^{\upsilon_T}(T|\mathcal{S}_T) \leq \frac{1}{\lambda_i(\upsilon_T)} \max_{S^j_{T'} \in \mathcal{S}^{\upsilon_T}} h_{ji}(T-T')\mathbb{1}_{X_{T'}\neq 0}. \tag{4.3.23}$$

Then we use the bounds for $\psi_i^{\upsilon_T}(T|\mathcal{S}_T)$ to build bounds for $\mathbb{P}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T)$ by discussing cases on the refractory period as we did in previous section:

$$\mathbb{P}_{min}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T) \leq \mathbb{P}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T) \leq \mathbb{P}_{max}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T). \tag{4.3.24}$$

If this range of probability $[\mathbb{P}_{min}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T), \mathbb{P}_{max}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T)]$ is small enough it allows to determine the thinning mark.

$$X_T = \begin{cases} 1 \quad \text{if} \quad U_T \leq \mathbb{P}_{min}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T) \\ 0 \quad \text{if} \quad U_T > \mathbb{P}_{max}(X_T = 1|V_T = \upsilon_T, \mathcal{F}_T) \\ n.a \quad \text{otherwise.} \end{cases} \tag{4.3.25}$$

Remark 36. Note that, in our setting, $\mathbb{P}_{min}$ stay to be 0 until one point in υ_T is said to be accepted. In this case, $\mathbb{P}_{min} = \mathbb{P}_{max} = \mathbb{P}$.

Pseudo-code

Algorithm 7 Algorithm KOtpP (for Kalikow-Ogata's thinning partially parallelized)

▷ **Step1: Seeding step**
for each *population* k **in parallel do**
  for each *neuron* $i \in$ population **do**
    Compute the number of *points* : n_i follows a $Poisson(M_i t_{max})$.
    for each *point* indexed by $k \leq n_i$ **do**
      Simulate the time : T_l^i follows a $U([0, t_{max}])$ distribution
      Define the uniform : $U_{T_l^i}$ follows a $U([0, 1])$ distribution
▷ **Step2: Kalikow step and pre-thinning**
      Pick neighborhood in space : $\{j\}$ wp $\lambda_i^{space}(\{j\})$
      Compute the upper bound $M_{v_{T_l^i}}$ which only depends on j
      if $U_{T_l^i} > \frac{M_{v_{T_l^i}}}{M_i}$ **then**
        Reject the *point* : the thinning mark $X_{T_l^i} = 0$
      else
        Set the *point* as unknown : the thinning mark $X_{T_l^i} = n.a$
        Pick neighborhood in time : $[T_l^i - (n+1)\delta, T_l^i - n\delta]$ wp $\lambda_i^{time}(\{n\})$
      end if
    end for
  end for
  Define $\mathcal{S}_{pop}^k = \bigcup \mathcal{S}^i$ the whole set of *points* belonging to the *population*
  Sort this set of *points* $\mathcal{S}_{pop}^k = \bigcup \mathcal{S}^i$ in an increasing T-order
end for
Define the whole set of *points* : $\mathcal{S} = \bigcup \mathcal{S}_{pop}^k$
▷ **Step3 : Thinning**
while At least one *point* $S_T^i \in \mathcal{S}$ is unknown **do**
  for each *population* k **in parallel do**
    for each unknown *point* $S_T^i \in \mathcal{S}_{pop}^k$ **do**
      Compute the upper bound probability $\mathbb{P}_{max}(X_T = 1 | V_T = v_T, \mathcal{F}_T)$ (4.3.24)
      Compute the lower bound probability $\mathbb{P}_{min}(X_T = 1 | V_T = v_T, \mathcal{F}_T)$ (4.3.24)
      Determine the thinning mark X_T if it is possible (4.3.25)
    end for
  end for
  for each *population* k **in sequence do**
    Update the whole set of *points* $\mathcal{S}$ with the thinning marks
  end for
end while
▷ **Final step : format the spikes trains**
Delete the points that have a thinning mark $X_T = 0$

Remark 37. 1. With more general choices of neighborhood, it suffices to accept/reject all unknown points in neighborhood υ_T to construct $\mathbb{P}_{min}, \mathbb{P}_{max}$, as long as the intensities are monotone with respect to $\mathcal{S}_T$.

2. In addition, with more general choices of neighborhood, it suffices to replace the Step 2 by choosing a neighborhood $\upsilon_{T_l^i}$ by Kalikow decomposition and do pre-thinning for each T_l^i.

3. Again, the crucial hypothesis in this section is the existence of $\mathbb{P}_{min}$ and $\mathbb{P}_{max}$. This is guaranteed as soon the intensity restricted to any neighborhood υ_T is increasing with respect to the configuration of thinning marks in neighborhood υ_T, that is

$$\phi^{\upsilon_T}(T|\mathcal{S}_T^{\upsilon_T}) \leq \phi^{\upsilon_T}(T|\mathcal{S}_T'^{\upsilon_T})$$

whenever $\mathcal{S}_T^{\upsilon_T} \preceq \mathcal{S}_T'^{\upsilon_T}$.

4.3.5 Applications

In this section, we explain how we concretely apply the algorithms to a multivariate linear Hawkes process with a hard refractory period δ, where the intensity of index i is recalled below.

$$\phi_i(t|\mathcal{F}_t) = \psi_i(t|\mathcal{F}_t)\mathbb{1}_{t-L_{Z^i}(t)>\delta} = \left(\mu_i + \sum_{j\in\mathbf{I}}\int_0^t h_{ji}(t-s)dZ_s^j\right)\mathbb{1}_{t-L_{Z^i}(t)>\delta}. \tag{4.3.26}$$

Settings

An interaction function h_{ji} is an exponentially decreasing function multiplied by a constant β_{ji} which depends on the distance $|i-j|$ between the two *neurons*.

$$h_{ji}(t) = \beta_{ji}\exp(-\alpha t) \tag{4.3.27}$$

with β_{ji} as follows

$$\beta_{ji} = \begin{cases} \dfrac{1}{2|j-i|^p} & \text{if } j \neq i \\ 1 \quad \text{if } j = i. \end{cases} \tag{4.3.28}$$

Upper bounds intensities for Ogata's thinning algorithm

The computation of the upper bound M_i is the same as pointed out in Lemma 1 of Section 4.6 of [18] or Chapter 3. We recall that $M_i = \mu_i + \dfrac{\sum_j \beta_{ji}}{1-\exp(-\alpha\delta)}$ the dominating intensity for index i.

Upper bounds intensities for Kalikow-Ogata's thinning algorithm

We prove the existence of M_υ and M_i for a multivariate linear Hawkes process with the hard refractory period. The following calculations and proofs are mostly based on [18] or Chapter 3. Since it is sufficient to do the calculus at a time T arbitrary, hence to simplify, we omit the subscript T at the notations. We consider the family of neighborhood $\mathbf{V}$ with $v = \{j\} \times [T-(n+1)\delta, T-n\delta)$ for $j \in \mathbf{I}$ and $v_\emptyset = \emptyset$. Following the method in [18], it admits a Kalikow decomposition at time T, for $v = \{j\} \times [T-(n+1)\delta, T-n\delta)$

$$\begin{cases} \lambda_i(v) = \lambda_{ji} > 0 \\ \phi_i^v(T|\mathcal{F}_T) = \dfrac{\int_{T-(n+1)\delta}^{T-n\delta} h_{ji}(T-s)dZ_s^j}{\lambda_{ji}} \mathbb{1}_{T-L_{Z^i}(T)>\delta}. \end{cases} \tag{4.3.29}$$

Otherwise, if $v = \emptyset$, we have

$$\begin{cases} \lambda_i(v) = \lambda_i(\emptyset) > 0 \\ \phi_i^v(T|\mathcal{F}_T) = \dfrac{\mu_i}{\lambda_i(\emptyset)} \mathbb{1}_{T-L_{Z^i}(T)>\delta} \end{cases} \tag{4.3.30}$$

with any sequence $(\lambda_{ji})_j$ such that $\sum_j \lambda_{ji} = 1 - \lambda_i(\emptyset)$.

Moreover, we also choose

$$\lambda_{ji} = \frac{C}{2|j-i|^p} = C\beta_{ji}$$

with C is a renormalization constant.

Also from Lemma 1 in Section 4.6 of [18] we can give the dominating intensities M_i and M_v as follow: for $v = \{j\} \times [T-(n+1)\delta, T-n\delta)$, we have

$$M_v = \frac{e^{-n\delta}}{C}.$$

Otherwise, if $v = \emptyset$ we have

$$M_\emptyset = \frac{\mu_i}{\lambda_i(\emptyset)}.$$

Therefore, we can choose

$$M_i = \sup_v M_v = \max\left\{ \frac{\mu_i}{\lambda_i(\emptyset)}, \frac{e^{-\delta}}{C} \right\}.$$

4.4 Second part: Stochastically bounded intensities

Most of the time, finding a constant that dominates the intensity of a neuron for the whole time of simulation is impossible, for example, the Linear Hawkes process [10], etc. However, as stated earlier, it remains possible to construct a dominating intensity, which changes over time and dominates the intensity of every neuron as we go along, see Assumption 8. This idea is not new, it can be found in [16]. This dominating intensity is updated point after point, so we construct it gradually.

For the simulation purposes, recall that the point processes we consider have empty past, i.e no point before time 0. To simplify the writing, throughout this section, we set $tmax = \infty$ and the goal is to simulate the first n_0 (deterministic) spikes. We focus on the two following versions of Linear Hawkes process.

$$\text{[Linear Hawkes process]} \qquad \phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} \int_0^t h_{ji}(t-s)dZ_s^j \tag{4.4.1}$$

and

$$\text{[Linear Hawkes process with bounded support } A > 0\text{]} \qquad \phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} \int_{t-A}^t h_{ji}(t-s)dZ_s^j. \tag{4.4.2}$$

For the classical version of the Linear Hawkes process, we show later that in some cases, the dominating intensity $\Lambda_i(t)$ might be a non-decreasing simple predictable function [22]. In other words, the dominating intensity may explode in finite time. This explains why we propose to stop the algorithm after obtaining n_0 spikes.

For the Linear Hawkes process with bounded support, the dominating intensity $\Lambda_i(t)$ now can be decreased after a delay of time A.

With stochastically bounded intensity, the Ogata's algorithm [16] can be rewritten as follow

i. Seeding step: We construct the dominating intensity $\Lambda_i(t)$ gradually. For each point S_T^i, we update the dominating intensity to find a candidate point.

ii. Thinning step: It is selected or rejected via an exact computation of the intensity at the corresponding time.

4.4.1 Sequential Ogata's thinning algorithm

Overall description

In the following, recall that the dominating process Π_i correspond to index i and has intensity $\Lambda_i(t)$. Moreover, recall that $\Pi = (\Pi_i)_i$ and denote $\Pi(1)$ for the list of accepted points after the thinning steps. We consider the filtration $\bar{\mathscr{F}}$ such that $\Lambda_i(t)$ is adapted to $\bar{\mathscr{F}}_t$ for every i:

$$\bar{\mathscr{F}}_t = \bigvee_{j\in\mathbf{I}} \{(T, X_T) : T \in \Pi^j \cap [0,t)\} \tag{4.4.3}$$

with X_T is the thinning random variable corresponds to T. Because $\bar{\mathscr{F}}$ includes the accepted points, one can see $\bar{\mathscr{F}}$ as an extension of $\mathscr{F}$. In addition, with this extended filtration, the intensity of Linear Hawkes process can be written as

$$\phi_i(t|\bar{\mathscr{F}}_t) = \mu_i + \sum_{j\in\mathbf{I}} \sum_{T<t:T\in\Pi^j} h_{ji}(t-T)\mathbb{1}_{X_T=1} \tag{4.4.4}$$

which is $\bar{\mathscr{F}}_t$ predictable. Similarly for Linear Hawkes process with bounded support defined in (4.4.2). In addition, we define the first time where the system has n_0 points.

$$\tau = \inf\{t \text{ such that } \sum_{T\leq t, T\in\Pi} \mathbb{1}_{X_T=1} \geq n_0\}.$$

Note that the dominating process Π will be constructed on $[0,\tau]$, outside of this interval, its value is 0.

Similarly to the deterministically bounded intensity, the proposed algorithm consists of two steps: seeding step and thinning step.

1. At the seeding step, we create a candidate to find the next point after time T. At this stage, we compute the dominating intensity $M(T) = \sum_{i=1}^N M_i(T)$ at time T, that creates a candidate T' by adding an exponential random variable of parameter $M(T)$. We attribute a mark to this received candidate. This candidate T' belongs to neuron i with probability $M_i(T)/M(T)$.

2. At the thinning step, assuming that it belongs to neuron i, we then assign to it a thinning mark with Bernoulli distribution of parameter $\mathbb{P}(X_{T'} = 1|\bar{\mathscr{F}}_{T'}) := \phi_i(T'|\bar{\mathscr{F}}_{T'})/M_i(T)$.

Remark 38. Note that, in order to find the next spike after time T, we use $M_i(T)/M(T)$ and $\phi_i(T'|\bar{\mathscr{F}}_{T'})/M_i(T)$. They are $\bar{\mathscr{F}}_T$ measurable, in other words, we only need the information up to time T and T included.

The thinning rate at the candidate point of time T' is given as follow:

$$X_{T'} = \begin{cases} 1 & \text{if} \quad U_{T'} \leq \mathbb{P}(X_{T'} = 1|\bar{\mathscr{F}}_{T'}) \\ 0 & \text{otherwise.} \end{cases} \tag{4.4.5}$$

Remark 39.
1. In comparison to previous section, where we start by simulating a Poisson process on each neuron, here the dominating intensity needs to be constructed gradually, point after point. This prevents us to do parallel computation.

2. Note that, this algorithm is a modification of the original paper of Ogata [16]. In the original paper, Ogata do labelling task. More precisely, at each candidate point, he assigns a mark from 0 to N where 0 means rejection. To do this task, we need to compute the convolutional sum of the intensities. This idea also appeared in [14]. However, when the number of neuron is huge, compute the convolutional sum of the intensities is very expensive. Whereas, in our algorithm, we only need to compute $M_i(T)$ for all i.

Pseudo-code

Algorithm 8 Algorithm OtSS for Ogata thinning sequential with stochastically bounded intensity

Initialize $T = 0$.
Initialize the number of spikes $n = 0$.
while $n \leq n_0$ **do**
$\triangleright$ **Step 1 : Seeding**
  Compute the dominating intensity $M(T) = \sum_i M_i(T)$, where $M_i(T)$ is defined in Assumption 8.
  Draw a candidate $T' \leftarrow T + Exp(M(T))$
  Distribute to T' a mark i with probability $M_i(T)/M(T)$
$\triangleright$ **Step 2: Thinning**
  Assume that, this candidate belongs to neuron i.
  Compute the thinning rate $\mathbb{P}(X_{T'} = 1|\bar{\mathcal{F}}_{T'}) = \dfrac{\phi_i(T'|\bar{\mathcal{F}}_{T'})}{M_i(T)}$
  Determine the thinning mark $X_{T'}$ by Equation (4.4.5). If accepted, increase n by 1.
  Update $T \leftarrow T'$
end while
$\triangleright$ **Final step : format the spikes trains**
Delete the points that have a thinning mark $X_T = 0$

Remark 40. 1. Note that, by the assumption of stochastically bounded intensity Assumption 8, for any $T' = T + Exp(M(T))$ a candidate after time T, we have

$$\phi_i(T'|\bar{\mathcal{F}}_{T'}) \leq M_i(T).$$

It means $\dfrac{\phi_i(T'|\bar{\mathcal{F}}_{T'})}{M_i(T)}$ is well defined as a probability.

By using Proposition 1 in [16], we can prove that the simulated points after Step 1 is a multivariate point process $\Pi = (\Pi^i)_i$ having conditional intensity $(\Lambda_i(t))_i$ up to stopping time τ. We can prove that

Proposition 10. Conditioning on the event $\{\tau < \infty\}$, the point process $\Pi^i(1)$ obtained points by Algorithm 9 admits $\phi_i(t|\bar{\mathcal{F}}_t)$ as $\bar{\mathcal{F}}_t$-predictable intensity up to the stopping time τ.

Proof. This proposition is a particular case of Proposition 11. We accept it for a moment. □

4.4.2 Sequential KalikowOgata algorithm

Overall description

For any neuron i, suppose that the intensity $\phi_i(t|\mathcal{F}_t)$ satisfies a Kalikow decomposition for all t. This means that, at each time t, there exists a neighborhood family $\mathbf{V}_t$, in which, we define a probability distribution $\lambda_i(v_t)$ and cylindrical functions $\phi_i^{v_t}(t|\mathcal{F}_t)$ such that

$$\phi_i(t|\mathcal{F}_t) = \sum_{v_t \in \mathbf{V}_t} \lambda_i(v_t)\phi_i^{v_t}(t|\mathcal{F}_t).$$

Assumption 10 (Stronger version of the stochastically bounded intensity)**.** There exists a dominating point process $\Pi = (\Pi^i)_i$ such that

1. At any T of the dominating process Π, there exists $M_i(T)$ a $\bar{\mathscr{F}}_T$ measurable random variable such that
$$\sup_{t>T} \sup_{v_t \in \mathbf{V}_t} \phi_i^{v_t}(t|\mathscr{F}_t) \mathbb{1}_{Z(T,t)=0} \leq M_i(T) < \infty. \tag{4.4.6}$$

2. For each index i, Π^i has $\bar{\mathscr{F}}$ intensity
$$\Lambda_i(t) = \sum_T M_i(T) \mathbb{1}_{T<t\leq n_\Pi(T)} \tag{4.4.7}$$

Note that, $\Lambda_i(t) \geq \phi_i(t|\mathscr{F}_t)$ almost surely and that therefore one might deduce points of Z by thinning points of Π and recall that $\bar{\mathscr{F}}$ is the extended filtration of $\mathscr{F}$ (4.4.3).

Intuitively, as soon as no point is produced, whatever the picked neighborhood v is, the intensity function restricted to this neighborhood is bounded by a predictable function. That allows us, at the same time, to construct the dominating intensity and do a thinning step.

Remark 41. Assumption 10 is stronger than Assumption 8. Indeed, from Assumption 10, we conclude that
$$\phi_i(t|\mathscr{F}_t) \mathbb{1}_{Z(T,t)=0} \leq M_i(T) \quad \forall i \in \mathbf{I}, \forall t > T, \forall T.$$

Analogous to Ogata's algorithm, we start with the seeding step, followed by a Kalikow step where we pick a random neighborhood. Finally, we do a thinning step to determine the marks. More precisely,

1. At the seeding step, we create a candidate to find the next point after time T. At this stage, we compute the dominating intensity $M(T) = \sum_{i=1}^N M_i(T)$ at time T, that creates a candidate T' by adding an exponential of parameter $M(T)$. We attribute the neuron mark to this received candidate T', this candidate belongs to neuron i with probability $M_i(T)/M(T)$. This is exactly step 1 of the previous algorithm

2. At the Kalikow step, by using the Kalikow decomposition, we pick a random neighborhood of T'. Without loss of generality, assume that $V_{T'} = v_{T'}$.

3. At the thinning step, we then assign to it a thinning mark with Bernoulli distribution of parameter $\mathbb{P}(X_{T'} = 1 | V_{T'} = v_{T'}, \bar{\mathscr{F}}_{T'}) = \phi_i^{v_{T'}}(T' | \bar{\mathscr{F}}_{T'})/M_i(T)$.

The thinning mark at the candidate T' is determined by
$$X_{T'} = \begin{cases} 1 & \text{if} \quad U_{T'} \leq \mathbb{P}(X_{T'} = 1 | V_{T'} = v_{T'}, \bar{\mathscr{F}}_{T'}) \\ 0 & \text{otherwise.} \end{cases} \tag{4.4.8}$$

PSEUDO-CODE

Algorithm 9 Algorithm KOtSS for KalikowOgata thinning sequential with stochastic bounded intensity

Initialize $T = 0$.
Initialize number of spikes $n = 0$.
while $n \leq n_0$ **do**
$\triangleright$ **Step 1: Seeding**
  Compute the dominating intensity of the candidate $M(T) = \sum_i M_i(T)$, where $M_i(T)$ is defined in Assumption 10.
  Draw a candidate $T' \leftarrow T + Exp(M(T))$
  Distribute to T' a mark i with probability $M_i(T)/M(T)$
$\triangleright$ **Step 2: Kalikow**
  Assume that, this candidate belongs to neuron i.
  Pick a neighborhood $V_{T'} = v_k$ with probability $\lambda_{i,T'}(v_k)$ in Kalikow decomposition. To simplify set $v_{T'} = v_k$.
$\triangleright$ **Step 3: Thinning**
  Compute the thinning rate $\mathbb{P}(X_{T'} = 1 | V_{T'} = v_{T'}, \bar{\mathcal{F}}_{T'})$
  Determine the thinning mark $X_{T'}$ by Equation (4.4.8). If accepted, increase n by 1.
  Update $T \leftarrow T'$
end while
$\triangleright$ **Final step : format the spikes trains**
Delete the points that have a thinning mark $X_T = 0$

By using Proposition 1 in [16], we can prove as before that the simulated points after Step 1 is a multivariate point process $\Pi = (\Pi^i)_i$ having conditional intensity $(\Lambda_i(t))_i$ respectively up to time τ. We can prove that

Proposition 11. Conditioning on the event $\{\tau < \infty\}$, the point process $\Pi^i(1)$ obtained by Algorithm 9 admits $\phi_i(t|\bar{\mathcal{F}}_t)$ as $\bar{\mathcal{F}}_t$-predictable intensity up to the stopping time τ.

Proof. The proof is similar to Proposition 2. Take C_t a positive, predictable function with respect to $\bar{\mathcal{F}}_t$. Let us denote T with $j_T = i$ by T_k^i for some k. We have

$$\mathbb{E}\left(\int_0^\tau C_t d\Pi_t^i(1)\right) = \sum_{k \in \mathbb{N}} \mathbb{E}\left(C_{T_k^i} \mathbb{1}_{X_{T_k^i}=1} \mathbb{1}_{T_k^i \leq \tau}\right).$$

Conditioning on $\{V_{T_k^i}, \bar{\mathcal{F}}_{T_k^i}\}$, we have

$$\begin{aligned}\mathbb{E}\left(\int_0^\tau C_t d\Pi_t^i(1)\right) &= \sum_k \mathbb{E}\left(C_{T_k^i} \mathbb{E}\left(\mathbb{1}_{X_{T_k^i}=1} \mathbb{1}_{T_k^i \leq \tau} | V_{T_k^i}, \bar{\mathcal{F}}_{T_k^i}\right)\right) \\ &= \sum_k \mathbb{E}\left(C_{T_k^i} \frac{\phi_i^{V_{T_k^i}}(T_k^i | \bar{\mathcal{F}}_{T_k^i})}{M_i(L_\Pi(T_k^i))} \mathbb{1}_{T_k^i \leq \tau}\right).\end{aligned}$$

Note that $\phi^v(.|\bar{\mathcal{F}}_t)$ is the cylindrical function that only depends on the points on the neighborhood v.

Let us now integrate with respect to the choice $V_{T_k^i}$, which is independent of anything else. Since for any T, $\Lambda_i(T) = M_i(L_\Pi(T))$, we conclude that

$$\mathbb{E}\left(\int_0^\tau C_t d\Pi_t^i(1)\right) = \sum_k \mathbb{E}\left(C_{T_k^i} \frac{\sum_{v_{T_k^i} \in \mathbf{V}_{T_k^i}} \lambda_i(v_{T_k^i}) \phi_i^{v_{T_k^i}}(T_k^i | \bar{\mathscr{F}}_{T_k^i})}{\Lambda_i(T_k^i)} \mathbb{1}_{T_k^i \leq \tau} \right)$$
$$= \mathbb{E}\left(\int_0^\tau C_t \frac{\phi_i(t|\bar{\mathscr{F}}_t)}{\Lambda_i(t)} d\Pi_t^i\right).$$

by definition of Kalikow decomposition for $\phi_i(t|\bar{\mathscr{F}}_t)$ and the fact that Π^i is empty outside the interval $[0,\tau]$. Moreover, since Π_t^i is constructed as point process with $\bar{\mathscr{F}}_t$ intensity $\Lambda_i(t)$, it leads us to conclude that

$$\mathbb{E}\left(\int_0^\tau C_t d\Pi_t^i(1)\right) = \mathbb{E}\left(\int_0^\tau C_t \phi_i(t|\bar{\mathscr{F}}_t) dt\right).$$

This holds for any C_t that is $\bar{\mathscr{F}}_t$ predictable. Therefore, it implies that $\phi_i(t|\bar{\mathscr{F}}_t)$ is the $\bar{\mathscr{F}}_t$ intensity of $\Pi^i(1)$. In other words, the obtaining points from Algorithm 9 admits $\phi_i(t|\bar{\mathscr{F}}_t)$ as stochastic intensity. This completes our proof. □

Remark 42. To prove Proposition 10, it is then sufficient to consider the Kalikow decomposition in a particular case, where the neighborhood V_T takes the whole past before time T with probability 1. Indeed, Ogata's algorithm is a particular case of the KalikowOgata algorithm by considering the neighborhood family $\mathbf{V}_T$ having only one element: the whole past before time T.

4.4.3 Applications

To illustrate our simulation algorithms, in this section, we consider Linear Hawkes processes [10, 11] and the Linear Hawkes processes with a bounded support [9]. Recall that the intensities are given as follow:

$$\text{[Linear Hawkes process]} \qquad \phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} \int_0^t h_{ji}(t-s) dZ_s^j \tag{4.4.9}$$

and

$$\text{[Linear Hawkes process with a bounded support]} \qquad \phi_i(t|\mathscr{F}_t) = \mu_i + \sum_{j\in\mathbf{I}} \int_{t-A}^t h_{ji}(t-s) dZ_s^j \tag{4.4.10}$$

where A is a positive number.

Settings

We consider the parameters in very general settings, that is

- μ_i a positive number, provides the intensity of the Hawkes process i when it does not interact with any process in the network.

- Each interaction function $(h_{ji}) : \mathbb{R}^+ \mapsto \mathbb{R}^+$ continuous and non increasing function, that represents the strength of the synaptic connection between the pre-synaptic process j and the post-synaptic process i [21].

Stochastic upper bounds for the intensities in Ogata's thinning algorithm

For any T and i, we recall the definition of upper stochastical bound $M_i(T)$ in Assumption 8,

$$\sup_{t>T} \phi_i(t|\mathcal{F}_t)\mathbb{1}_{Z(T,t)=0} \leq M_i(T).$$

In the following of this example (and only in this example), we will write h_{ji} as h^i_j to remove any unexpected confusion to the notations.

With the Linear Hawkes process (4.4.1), we split the integral into two parts

$$\sup_{t>T} \phi_i(t|\mathcal{F}_t)\mathbb{1}_{Z(T,t)=0} = \mu_i + \sup_{t>T}\left(\sum_j \int_0^T h^i_j(t-s)dZ^j_s + h^i_{j_T}(t-T)\mathbb{1}_{X_T=1}\right).$$

Finally, since h^i_j is non increasing function, we set

$$M_i(T) = \mu_i + Z((0,T)) \max_j h^i_j(0) + h^i_{j_T}(0)\mathbb{1}_{X_T=1}.$$

In which, we recall that $Z((a,b))$ counts the number of spikes of the process Z in the interval (a,b). Moreover, $\Lambda_i(t) = \sum_T M_i(T)\mathbb{1}_{T<t\leq n_\Pi(T)}$ is $\mathcal{F}^\Pi_t$ predictable, non decreasing function where recall that $n_\Pi(T)$ is the next point after time T of the process Π. Indeed, denote $T' = n_\Pi(T)$, we have

$$M_i(T) = \mu_i + Z((0,T)) \max_j h^i_j(0) + h^i_{j_T}(0)\mathbb{1}_{X_T=1} \leq \mu_i + Z((0,T')) \max_j h^i_j(0) + h^i_{j_{T'}}(0)\mathbb{1}_{X_{T'}=1} = M_i(T').$$

We also can construct a more precise dominating intensity that decreases after each rejected point by considering

$$M_i(T) = \mu_i + \sum_j \int_0^T h^i_j(T-s)dZ^j_s + h^i_{j_T}(0)\mathbb{1}_{X_T=1}.$$

That is also $\mathcal{F}^\Pi_T$ measurable. Moreover, if $T' = n_\Pi(T)$ is the candidate point after T and $X_{T'} = 0$, we have

$$M_i(T') = \mu_i + \sum_j \int_0^T h^i_j(T'-s)dZ^j_s + h^i_{j_T}(0)\mathbb{1}_{X_T=1} < M_i(T)$$

since h^i_j is a non-increasing function.

We obtain a similar result with the Linear Hawkes process with bounded support (4.4.10),

$$M_i(T) = \mu_i + Z([T-A,T)) \max_j h^i_j(0) + h^i_{j_T}(0)\mathbb{1}_{X_T=1}.$$

Clearly, for each T' rejected and until we meet a spike, the dominating intensity will decrease after delays of length A.

Stochastic upper bounds for the intensities in KalikowOgata's thinning algorithm

Firstly, we begin to write Kalikow decomposition for the intensity function $\phi_i(t|\mathscr{F}_t)$. Let us consider the neighborhood family $\mathbf{V}_t$ with $v_k = \{k\}\times[0,t)$ for $k \in \mathbf{I}$. Follow the method in [18], the intensity of Linear Hawkes process admits a Kalikow decomposition at time t for $k > 0$:

$$\begin{cases} \lambda_i\big(v_k\big) = \lambda_{ki} > 0 \\ \phi_i^{v_k}(t|\mathscr{F}_t) = \dfrac{\int_0^t h_{ki}(t-s)dZ_s^k}{\lambda_{ki}} \end{cases} \tag{4.4.11}$$

and for $k = 0$, $v_0 = \varnothing$:

$$\begin{cases} \lambda_i(\varnothing) = \lambda_{0i} > 0 \\ \phi_i^{\varnothing}(t) = \dfrac{\mu_i}{\lambda_{0i}}. \end{cases} \tag{4.4.12}$$

Then, at each T, by Assumption 10,

$$\sup_{t>T}\sup_{k} \phi_i^{v_k}(t|\mathscr{F}_t)\mathbb{1}_{Z((T,t))=0} \leq M_i(T).$$

We set $M_i(T) = \sup\bigg(\sup_k\bigg(\dfrac{1}{\lambda_{ki}}\big(Z^k((0,T)) + 1\big)h_{ki}(0)\bigg), \dfrac{\mu_i}{\lambda_{0i}}\bigg)$.

For the Linear Hawkes process with bounded support, the result is very similar. We have Kalikow decomposition at time t for $k > 0$:

$$\begin{cases} \lambda_i\big(v_k\big) = \lambda_{ki} > 0 \\ \phi_i^{v_k}(t|\mathscr{F}_t) = \dfrac{\int_{t-A}^t h_{ki}(t-s)dZ_s^k}{\lambda_{ki}} \end{cases} \tag{4.4.13}$$

and for $k = 0$, $v_0 = \varnothing$:

$$\begin{cases} \lambda_i(\varnothing) = \lambda_{0i} > 0 \\ \phi_i^{\varnothing}(t) = \dfrac{\mu_i}{\lambda_{0i}}. \end{cases} \tag{4.4.14}$$

We conclude that

$$M_i(T) = sup\bigg(\sup_k\bigg(\frac{1}{\lambda_{ki}}\big(Z^k((T-A,T)) + 1\big)h_{ki}(0)\bigg), \frac{\mu_i}{\lambda_{0i}}\bigg).$$

4.5 Numerical results

In this section, we prove statistically that the algorithms in Section 4.3 return Hawkes processes with the right intensity and assert that the new algorithms outperform the classical Ogata's algorithm [16]. The simulation is performed with a laptop computer with a processor Intel(R) Core(TM) i7-1165G7, base frequency 2,80 GHz, RAM 16 Go, and 4 hearts. The algorithms were implemented in Python programming language (version 3.8.3). The algorithms were run in parallel by using package multiprocessing in Python (version 3.8.3). The plots and statistical analysis were obtained by using Python (version 3.8.3), part of it using library *statsmodels* (v0.13.2).

4.5.1 Statistical test

In 1988, Ogata [17] introduced a standard method to assess if the data obey a point process with a given intensity, in particular Hawkes processes. This method is based on the time rescaling theorem (see for instance [2]), which says that if $\lambda(.)$ is the conditional intensity of the point process Z and $\Lambda(t) = \int_0^t \lambda_s ds$, then the point process $\tilde{Z} = \{\Lambda(T), T \in Z\}$ is the homogeneous Poisson process with rate 1. Ogata then derived a method to test a point process Z has a given intensity $\lambda(.)$ as follows:

- Applying the transform formula Λ to each arrival time of Z to obtain $\tilde{Z}$.
- Test 1: Test that the consecutive delay between points of $\tilde{Z}$ obeys the exponential distribution of rate 1, by using the Kolmogorov-Smirnov test.
- Test 2: Test that the points of $\tilde{Z}$ are uniformly distributed by using the Kolmogorov-Smirnov test.
- Test 3: Test that the delays between points of $\tilde{Z}$ are independent, for instance, we perform an auto-correlation test between delays with a certain lag, here we choose up to lag 9.

We perform a test with the 200 neurons that are simulated during 5s (i.e $t_{max} = 5$). We consider the Hawkes process with a hard refractory period and bounded support with intensity is given by:

$$\phi_i(t|\mathcal{F}_t) = \left(\mu_i + \sum_{j\in \mathbf{I}} \int_{t-A}^{t} h_{ji}(t-s)dZ_s^j\right)\mathbb{1}_{t-L_{Z^i(t)}>\delta}.$$

with $\mu_i = 1\forall i$, $A = 0.1$, $\delta = 0.01$, $\mathbf{I} = \{1, 2, \dots, 200\}$ and h_{ji} is defined in (4.3.27) and (4.3.28) with $\alpha = 2, p = 6$.

If we have indeed simulated a right Hawkes processes, the p values should be uniformly distributed in Test 1 and Test 2. In addition, in Test 3, we choose 8 neurons randomly. The auto-correlation function should be around 0 if the delays between points are independent.

Remark 43. The OtpP algorithm is extremely slow, even worse than OtS, hence it is out of interest in practice. Therefore, we will not present its result in this section.

Algorithm OtS

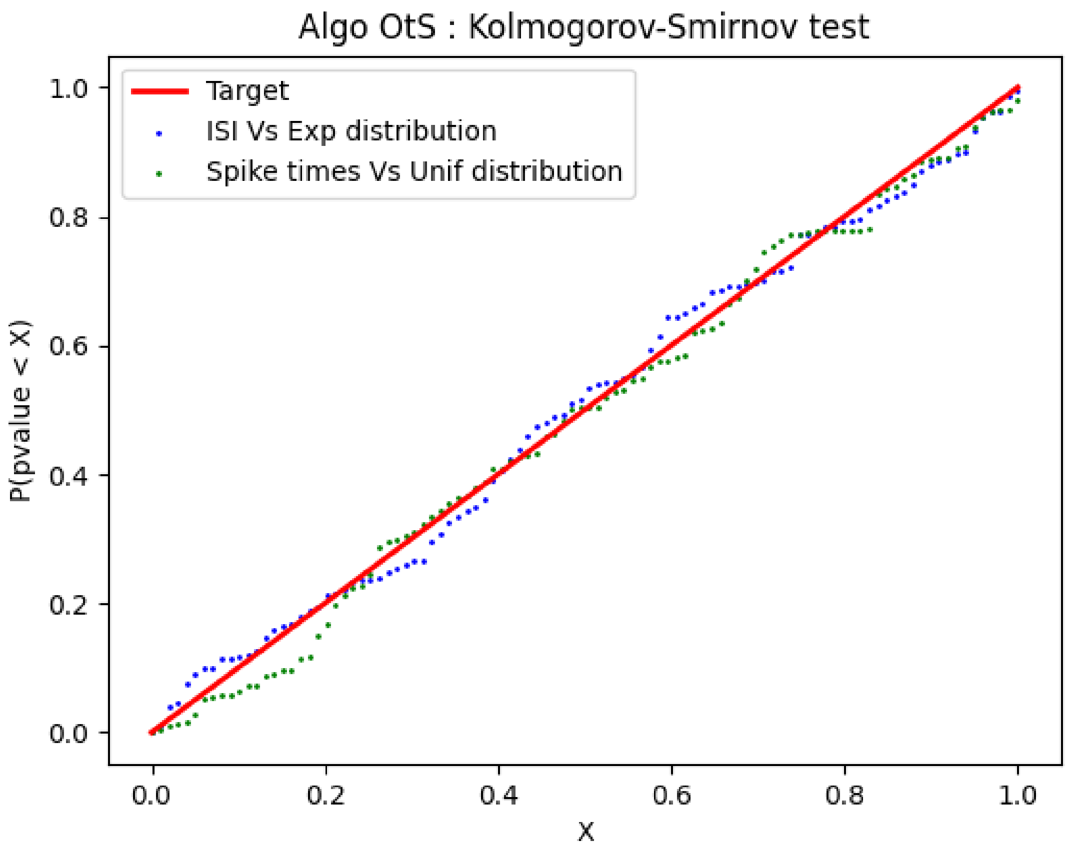


Figure 4.1: The cumulative distribution function (c.d.f) of the p values of Test 1 and Test 2 for OtS algorithm. The red diagonal line is the c.d.f of uniform random variable on [0,1]. The dashed blue line is the c.d.f of p values corresponds to Test 1. The dashed green line is the c.d.f of p values corresponds to Test 2.

On Figure 4.1, we can observe that the dashed blue line and green line is close to the red line. This means that the p values of Test 1 and Test 2 are uniformly distributed.

On Figure 4.2, we can observe that the vertical lines stay on the blue shaded region. This means that there is no significant correlation between the delays.

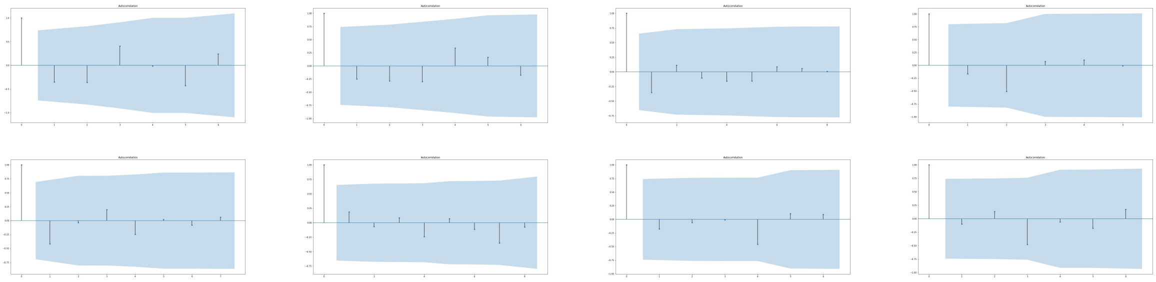


Figure 4.2: On each subfigure, the x-axis displays the number of lags and the y-axis displays the auto-correlation at that number of lags. By default, the plot starts at lag = 0 and the auto-correlation will always be 1 at lag = 0. The blue shaded region is the confidence interval with a default value of $\alpha = 0.05$.

Algorithm KOtS

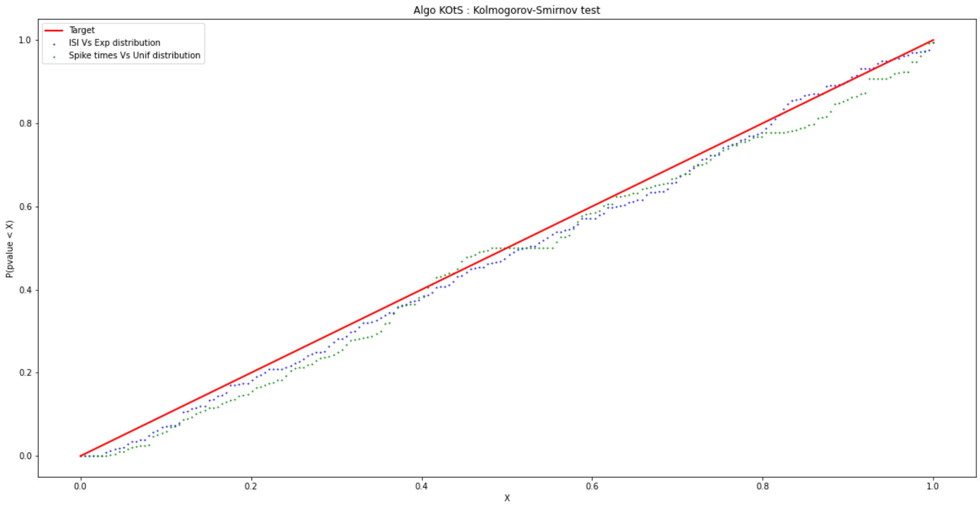


Figure 4.3: The c.d.f of the p values of of Test 1 and Test 2. The red diagonal line is the c.d.f of uniform random variable on [0,1]. The dashed blue line is the c.d.f of p values corresponds to Test 1. The dashed green line is the c.d.f of p values corresponds to Test 2.

On Figure 4.3, we can observe that the dashed blue line and green line is close to the red line. This means that the p values of Test 1 and Test 2 are uniformly distributed.

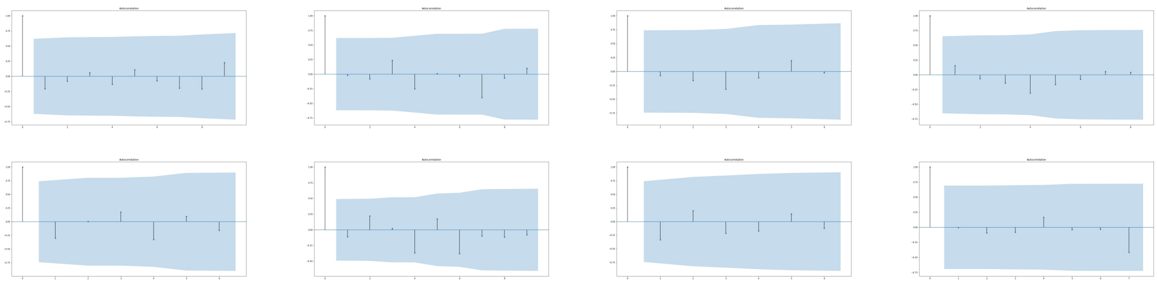


Figure 4.4: On each subfigure, the x-axis displays the number of lags and the y-axis displays the auto-correlation at that number of lags. By default, the plot starts at lag = 0 and the auto-correlation will always be 1 at lag = 0. The blue shaded region is the confidence interval with a default value of $\alpha = 0.05$.

On Figure 4.4, we can observe that the vertical lines stay in the blue shaded region. This means that there is no significant correlation between the delays.

Algorithm KOtpP

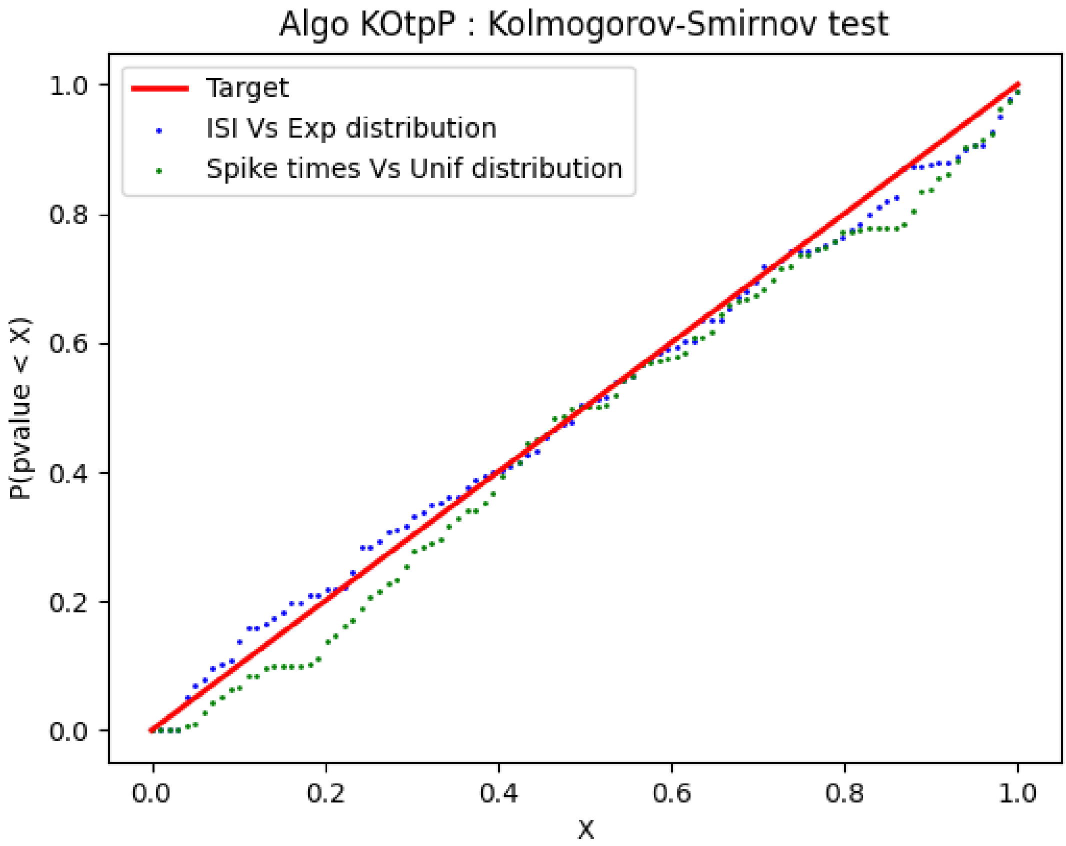


Figure 4.5: The c.d.f of the p values of of Test 1 and Test 2. The red diagonal line is the c.d.f of uniform random variable on [0,1]. The dashed blue line is the c.d.f of p values corresponds to Test 1. The dashed green line is the c.d.f of p values corresponds to Test 2.

On Figure 4.5, we can observe that the dashed blue line and green line is close to the red line. This means that the p values of Test 1 and Test 2 are uniformly distributed. On Figure 4.6, we can observe that the vertical lines stay in the blue shaded region. This means that there is no significant correlation between the delays.

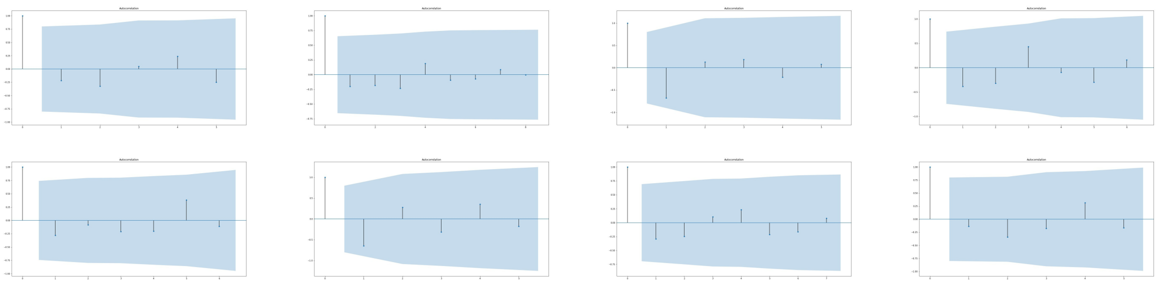


Figure 4.6: On each subfigure, the x-axis displays the number of lags and the y-axis displays the auto-correlation at that number of lags. By default, the plot starts at lag = 0 and the auto-correlation will always be 1 at lag = 0. The blue shaded region is the confidence interval with a default value of $\alpha = 0.05$.

4.5.2 Execution time of the algorithms

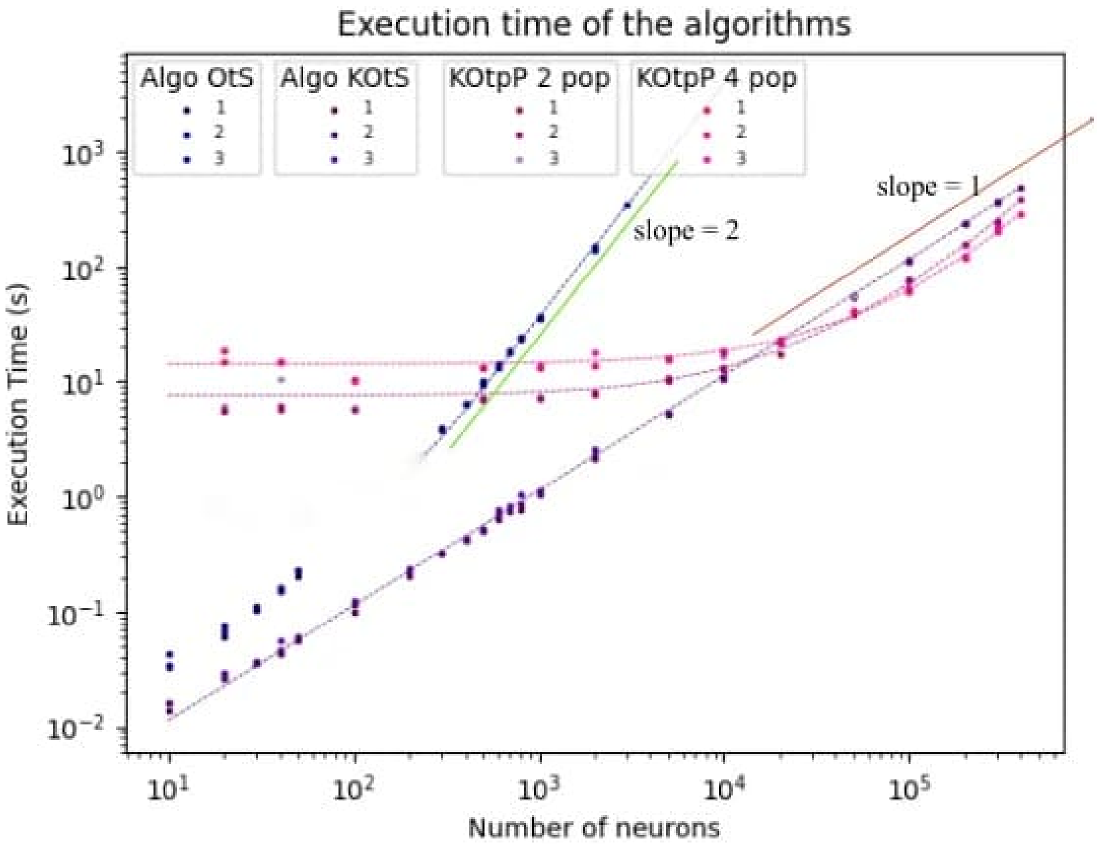


Figure 4.7: Execution time of the algorithms sequential Ogata's thinning, sequential Kalikow-Ogata's thinning, and partially parallel Kalikow-Ogata's thinning with 2 and 4 *poulations*: For each number of neurons, the algorithms are run 3 times. They simulate the process during 5s. The median values are used to perform a polynomial regression of degree 2

.

With the Algorithm OtS we reach the simulation of 3 000 *neurons* in 342.2 seconds (median). The performances of the algorithms based on the Kalikow decomposition are much better with 400 000 *neurons* sequentially simulated in 494.6 seconds (median). Using parallel computing, the algorithm KOtpP reaches the same number of *neurons* in 384.3 seconds (median) with 2 *populations* and 289.25 seconds (median) with 4 *populations*. Parallel computation with 4 *populations* decreases the execution time by 41.5% for simulating 400 000 *neurons*. In general, in the huge network (more than 20 000), the KOtpP performs significantly better than OtS, and better than KOtS.

However, with the number of *neurons* smaller than 20 000, the parallelization lost performances. Two reasons are noticed. The first one is the existence of a fixed cost in time with parallel computation. We assume that this is the synchronization time, it would probably be reduced with fewer alternations between sequential and parallel phases of computations. The other reason is the fact that for a small number of neurons it becomes more efficient to sort the whole set of *points*. Indeed we noticed that the smaller the number of *neurons*, the larger the number of *points* unknown after the first iteration. This can be explained by the fact that when the number of *neurons* is small the *populations* are small. Therefore *neurons* are more likely to have important interactions with *neurons* from other *populations*. In other words, the larger a *population*, the more independent it is. We can see it well for 4 *populations* and several *neurons* between 10 and 100. In this interval, the execution time decreases with respect to the number of *neurons*.

Remark 44. However, it is worth noting that, Python is not suitable for parallel computing. There are 2 paradigms of parallel computing: multithreading and multiprocessing. There exist a mechanism in Python called ”Global Interpreter lock” or GIL that prevents two or more thread from executing simultaneously. Therefore it is impossible to benefit from multithreading in Python. For multiprocessing, it is possible to run two or more Python processes in parallel, as each process has its interpreter. However, each process exists in its memory stack and is thus unable to exchange data unless an external synchronization mechanism (for example, MPI) is employed. To conclude, Paul Gresland and the engineers in NeuroMod and Inria (Sophia Antipolis) are working together to recode the algorithms in C++ to benefit the parallel computing.

References for Chapter 4

1. R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris, et al. “Simulation of networks of spiking neurons: a review of tools and strategies”. *Journal of computational neuroscience* 23:3, 2007, pp. 349–398.
2. E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank. “The time-rescaling theorem and its application to neural spike train data analysis”. *Neural computation* 14:2, 2002, pp. 325–346.
3. J. Chevallier. “Mean-field limit of generalized Hawkes processes”. *Stochastic Processes and their Applications* 127:12, 2017, pp. 3870–3912.
4. F. Comets, R. Fernández, and P. A. Ferrari. “Processes with long memory: regenerative construction and perfect simulation”. *The Annals of Applied Probability* 12:3, 2002, pp. 921–943.
5. R. Fernández, P. Ferrari, and A. Galves. “Coupling, renewal and perfect simulation of chains of infinite order”. *Lecture Notes for the vth Brazilian school of Probability, Ubatuba* 2001, 2001.
6. P. A. Ferrari, R. Fernández, and N. L. Garcia. “Perfect simulation for interacting point processes, loss networks and Ising models”. *Stochastic Processes and their Applications* 102:1, 2002, pp. 63–88.
7. A. Galves, N. Garcia, E. Löcherbach, and E. Orlandi. “Kalikow-type decomposition for multicolor infinite range particle systems”. *The Annals of Applied Probability* 23:4, 2013, pp. 1629–1659.
8. A. Galves and E. Löcherbach. “Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets”. *Journal of Statistical Physics* 151:5, 2013, pp. 896–921.
9. N. R. Hansen. “Hawkes processes and combinatorial transcriptional regulation”. PhD thesis. Citeseer, 1910.
10. A. G. Hawkes. “Point spectra of some mutually exciting point processes”. *Journal of the Royal Statistical Society: Series B (Methodological)* 33:3, 1971, pp. 438–443.

11. A. G. Hawkes. “Spectra of some self-exciting and mutually exciting point processes”. *Biometrika* 58:1, 1971, pp. 83–90.

12. P. Hodara and E. Löcherbach. “Hawkes processes with variable length memory and an infinite number of components”. *Advances in Applied Probability* 49:1, 2017, pp. 84–107.

13. S. Kalikow. “Random Markov processes and uniform martingales”. *Israel Journal of Mathematics* 71:1, 1990, pp. 33–54.

14. C. Mascart, A. Muzy, and P. Reynaud-Bouret. “Efficient Simulation of Sparse Graphs of Point Processes”. *arXiv preprint arXiv:2001.01702*, 2020.

15. C. Mascart, G. Scarella, P. Reynaud-Bouret, and A. Muzy. “Simulation scalability of large brain neuronal networks thanks to time asynchrony”, 2021.

16. Y. Ogata. “On Lewis’ simulation method for point processes”. *IEEE transactions on information theory* 27:1, 1981, pp. 23–31.

17. Y. Ogata. “Statistical models for earthquake occurrences and residual analysis for point processes”. *Journal of the American Statistical association* 83:401, 1988, pp. 9–27.

18. T. C. Phi. “Kalikow decomposition for counting processes with stochastic intensity”. *arXiv preprint arXiv:2104.00495*, 2021.

19. T. C. Phi, A. Muzy, and P. Reynaud-Bouret. “Event-scheduling algorithms with Kalikow decomposition for simulating potentially infinite neuronal networks”. *SN Computer Science* 1:1, 2020, pp. 1–10.

20. M. B. Raad and E. Löcherbach. “Stability for Hawkes processes with inhibition”. *Electronic Communications in Probability* 25:none, 2020, pp. 1–9. DOI: 10.1214/20-ECP312. URL: https://doi.org/10.1214/20-ECP312.

21. P. Reynaud-Bouret, V. Rivoirard, and C. Tuleau-Malot. “Inference of functional connectivity in neurosciences via Hawkes processes”. In: *2013 IEEE global conference on signal and information processing*. IEEE. 2013, pp. 317–320.

22. P. Tankov. *Financial modelling with jump processes*. Chapman and Hall/CRC, 2003.

23. H. Yamaura, J. Igarashi, and T. Yamazaki. “Simulation of a human-scale cerebellar network model on the k computer”. *Frontiers in neuroinformatics* 14, 2020, p. 16.

Bibliography

1. E. Bacry, I. Mastromatteo, and J.-F. Muzy. “Hawkes processes in finance”. *Market Microstructure and Liquidity* 1:01, 2015, p. 1550005.
2. P. Brémaud. *Point processes and queues: martingale dynamics*. Vol. 50. Springer, 1981.
3. P. Brémaud and L. Massoulié. “Stability of nonlinear Hawkes processes”. *The Annals of Probability*, 1996, pp. 1563–1588.
4. J. Chevallier. “Mean-field limit of generalized Hawkes processes”. *Stochastic Processes and their Applications* 127:12, 2017, pp. 3870–3912.
5. E. Choi, N. Du, R. Chen, L. Song, and J. Sun. “Constructing disease network and temporal progression model via context-sensitive hawkes process”. In: *2015 IEEE International Conference on Data Mining*. IEEE. 2015, pp. 721–726.
6. F. Comets, R. Fernández, and P. A. Ferrari. “Processes with long memory: regenerative construction and perfect simulation”. *The Annals of Applied Probability* 12:3, 2002, pp. 921–943.
7. A. Dassios and H. Zhao. “Exact simulation of Hawkes process with exponentially decaying intensity”. *Electronic Communications in Probability* 18, 2013, pp. 1–13.
8. P. A. Ferrari, R. Fernández, and N. L. Garcia. “Perfect simulation for interacting point processes, loss networks and Ising models”. *Stochastic Processes and their Applications* 102:1, 2002, pp. 63–88.
9. P. A. Ferrari, A. Maass, S. Martínez, and P. Ney. “Cesaro mean distribution of group automata starting from measures with summable decay”. *Ergodic Theory and Dynamical Systems* 20:6, 2000, pp. 1657–1670.
10. A. Galves, N. Garcia, E. Löcherbach, and E. Orlandi. “Kalikow-type decomposition for multicolor infinite range particle systems”. *The Annals of Applied Probability* 23:4, 2013, pp. 1629–1659.
11. A. Galves and E. Löcherbach. “Infinite systems of interacting chains with memory of variable length—a stochastic model for biological neural nets”. *Journal of Statistical Physics* 151:5, 2013, pp. 896–921.
12. A. G. Hawkes. “Point spectra of some mutually exciting point processes”. *Journal of the Royal Statistical Society: Series B (Methodological)* 33:3, 1971, pp. 438–443.
13. A. G. Hawkes. “Spectra of some self-exciting and mutually exciting point processes”. *Biometrika* 58:1, 1971, pp. 83–90.
14. P. Hodara and E. Löcherbach. “Hawkes processes with variable length memory and an infinite number of components”. *Advances in Applied Probability* 49:1, 2017, pp. 84–107.

15. P. Hodara and E. Löcherbach. “Hawkes processes with variable length memory and an infinite number of components”. *Adv. in Appl. Probab.* 49:1, 2017, pp. 84–107. ISSN: 0001-8678. DOI: 10.1017/apr.2016.80. URL: https://doi.org/10.1017/apr.2016.80.
16. S. Kalikow. “Random Markov processes and uniform martingales”. *Israel Journal of Mathematics* 71:1, 1990, pp. 33–54.
17. W. S. Kendall and J. Møller. “Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes”. *Advances in applied probability* 32:3, 2000, pp. 844–865.
18. P. W. Lewis and G. S. Shedler. “Simulation of nonhomogeneous Poisson processes by thinning”. *Naval research logistics quarterly* 26:3, 1979, pp. 403–413.
19. C. Mascart, A. Muzy, and P. Reynaud-Bouret. “Efficient Simulation of Sparse Graphs of Point Processes”. *arXiv preprint arXiv:2001.01702*, 2020.
20. J. Møller and J. G. Rasmussen. “Approximate simulation of Hawkes processes”. *Methodology and Computing in Applied Probability* 8:1, 2006, pp. 53–64.
21. J. Møller and J. G. Rasmussen. “Perfect simulation of Hawkes processes”. *Advances in applied probability* 37:3, 2005, pp. 629–646.
22. Y. Ogata. “On Lewis’ simulation method for point processes”. *IEEE Transactions on Information Theory* 27:1, 1981, pp. 23–31. DOI: 10.1109/TIT.1981.1056305.
23. Y. Ogata. “On Lewis’ simulation method for point processes”. *IEEE transactions on information theory* 27:1, 1981, pp. 23–31.
24. Y. Ogata. “Seismicity analysis through point-process modeling: A review”. *Seismicity patterns, their statistical significance and physical meaning*, 1999, pp. 471–507.
25. G. Ost and P. Reynaud-Bouret. “Sparse space–time models: Concentration inequalities and Lasso”. In: *Annales de l’Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 56. 4. Institut Henri Poincaré. 2020, pp. 2377–2405.
26. T. C. Phi. “Kalikow decomposition for counting processes with stochastic intensity”. *arXiv preprint arXiv:2104.00495*, 2021.
27. T. C. Phi, A. Muzy, and P. Reynaud-Bouret. “Event-scheduling algorithms with Kalikow decomposition for simulating potentially infinite neuronal networks”. *SN Computer Science* 1:1, 2020, pp. 1–10.
28. J. G. Propp and D. B. Wilson. “Exact sampling with coupled Markov chains and applications to statistical mechanics”. *Random Structures & Algorithms* 9:1-2, 1996, pp. 223–252.
29. M. B. Raad and E. Löcherbach. “Stability for Hawkes processes with inhibition”. *Electronic Communications in Probability* 25:none, 2020, pp. 1–9. DOI: 10.1214/20-ECP312. URL: https://doi.org/10.1214/20-ECP312.
30. J. G. Rasmussen. “Temporal point processes: the conditional intensity function”. *Lecture Notes, Jan*, 2011.

31. P. Reynaud-Bouret and S. Schbath. "Adaptive estimation for Hawkes processes; application to genome analysis". *The Annals of Statistics* 38:5, 2010, pp. 2781–2822.

32. L. Zhu. "Nonlinear Hawkes processes". PhD thesis. New York University, 2013.