
Mémoire de stage

Processus de Hawkes : application en épidémiologie végétale

Paul Pierrat

Master Ingénierie Mathématique pour la Science des données (IMSD)
Université de Lorraine

Réalisé à l'INRAE de Jouy-en-Josas

Avril-Septembre 2023

Encadré par **Katarzyna Adamczyk** (INRAE) et **Madalina Deaconu** (Inria)



1 Liste de mots-clés

Processus ponctuels, Processus de Poisson, Processus de Hawkes, Simulation, Inférence, Étude de données épidémiques.

2 Résumé

Dans ce travail, on étudie les processus de Poisson et de Hawkes univariés. Les propriétés des processus sont analysées et les démonstrations réalisées. Des algorithmes de simulation des processus sont étudiés ainsi que leur implémentation sous le logiciel **R**. Une méthode d'estimation des paramètres est appliquée par l'estimateur du maximum de vraisemblance et une démarche de validation de modèle par analyse résiduelle est exécutée.

Ces processus sont ensuite ajustés à des données épidémiques dans l'objectif de comprendre la dynamique du champignon *Venturia Inaequalis* responsable de la tavelure des pommiers. Après avoir réalisé une comparaison des modèles, le processus qui s'ajuste le mieux aux données est le processus de Hawkes.

Table des matières

1	Liste de mots-clés	3
2	Résumé	3
3	Introduction	6
3.1	Présentation de l'institut	6
3.2	Contexte biologique	6
3.3	Etat de l'art	8
3.4	L'objectif de stage	8
4	Processus de comptage et de Poisson	9
4.1	Définition et propriété	9
4.2	Processus de Poisson homogène	9
4.2.1	Simulation	17
4.2.2	Estimation des paramètres	17
4.3	Processus de Poisson non homogène	19
4.3.1	Simulation	20
4.3.2	Inférence	23
5	Processus de Hawkes	25
5.1	Intensité conditionnelle	25
5.2	Définitions et propriétés du processus de Hawkes	28
5.3	Simulation	32
5.3.1	Algorithme d'amincissement modifié d'Ogata	32
5.3.2	Simulation par l'inverse du compensateur	34
5.4	Inférence statistique	35
5.4.1	Vraisemblance du modèle	35
5.4.2	Validation de modèle	41
6	Modélisation des données épidémiques	41
6.1	Piégeage de spores et données météorologique	41
6.2	Ajustement du processus de Poisson non homogène avec une intensité log-linéaire	45
6.3	Ajustement du processus de Hawkes	48
6.4	Comparaisons des modèles	48
7	Conclusion	49
8	Annexe	53
8.1	Simulation du Processus de Poisson homogène	53
8.2	Simulation du processus de Poisson non homogène	53
8.3	Validation du modèle : Processus de Poisson non homogène	55
8.4	Simulation du processus de Hawkes	56
8.5	Recherche du maximum de vraisemblance Hawkes	58
8.6	Hessienne et normalité asymptotique Hawkes	61
8.7	Validation du modèle Hawkes	65

8.8	Application du processus de Hawkes à des données simulées	66
8.9	Application du processus de Hawkes aux données de libération de spores	66
8.10	Préparation des données	68
8.11	Ajustement au processus de Poisson non homogène avec une intensité log-linéaire . .	72

3 Introduction

3.1 Présentation de l'institut

L'institut national de recherche pour l'agriculture, l'alimentation et l'environnement **INRAE** est un établissement public à caractère scientifique et technologique (EPST) créé en 2020 suite à la fusion de l'INRA et l'IRSTEA. D'après son décret fondateur [NOR, 2019] : *"L'institut a pour missions de réaliser, d'organiser et de coordonner, à son initiative ou à la demande de l'Etat, tous travaux de recherche scientifique et technologique dans les domaines de l'agriculture, de l'alimentation, de la forêt, de l'environnement, de l'eau, de la biodiversité, de la bioéconomie, de l'économie circulaire, de la gestion durable des territoires et des risques"*. L'institut est structuré en 14 départements de recherche répartis en 19 centres de recherche.

J'ai réalisé mon stage dans le département de recherche "Mathématique et Numérique" (MATH-NUM) de l'INRAE, dans l'Unité de recherche "Mathématiques et Informatique Appliquées du Génome à l'Environnement" (MaIAGE¹), localisée dans le centre de recherche de Jouy-en-Josas. Cette unité *"regroupe des mathématiciens, des informaticiens, des bioinformaticiens et des biologistes autour de questions de biologie et agro-écologie, allant de l'échelle moléculaire à l'échelle du paysage en passant par l'étude de l'individu, de populations ou d'écosystèmes"*. J'ai travaillé dans l'équipe Dynenvie qui s'intéresse à la modélisation dynamique et statistique pour les écosystèmes, l'agronomie et l'épidémiologie.

Mon stage a été réalisé en collaboration avec l'équipe PASTA de l'**Inria** (Institut National de recherche en Informatique et Mathématiques appliquées). Cette équipe étudie les processus aléatoire spatio-temporels et leur applications.

3.2 Contexte biologique

Le stage a porté sur la modélisation de la propagation de la tavelure des pommiers ([INRAE, 2016]). La tavelure est une maladie des pommiers causée par un champignon appelé *Venturia Inaequalis*. L'infection peut altérer l'aspect de la pomme, mais peut aussi attaquer le fruit menant à une dégradation des récoltes (voir la Figure 1(a)).

La maladie est entièrement provoquée par la libération de petites particules de ce champignon appelées spores. Les spores sont contenues dans des cellules appelées asques (voir la Figure 1(b)). Lorsqu'elles sont éjectées, elles sont susceptibles de s'accrocher à des nouvelles feuilles saines ou fruits sains et ainsi les infecter. L'infection correspond à la germination de spores dans le nouveau fruit hôte. Cette étape, également appelée maturation des spores, permet à un nouveau champignon de se former. Ce dernier pourra également produire des spores et les libérer. De cette manière la maladie s'entretient et se développe (Figure 2). Les conditions climatiques et la variété de la pomme influent particulièrement sur le développement de la tavelure. Une chaleur trop élevée ou trop basse empêche la croissance des spores. La pluie et l'humidité des feuilles favorisent la libération de spores. En effet, lorsque de l'eau est déposée sur les feuilles ou les fruits infectés, la pellicule contenant les spores est retirée et les spores sont éjectées. De plus, il existe des variétés de pommes plus ou moins sensibles à la tavelure. Certaines variétés sont qualifiées de résistantes, mais cette résistance n'est pas durable.

En général, des années particulièrement pluvieuses mènent à de sévères dégradations des fruits. C'est pourquoi les agriculteurs utilisent des traitements préventifs qu'ils appliquent aux arbres pour

1. <https://maiage.inrae.fr/>

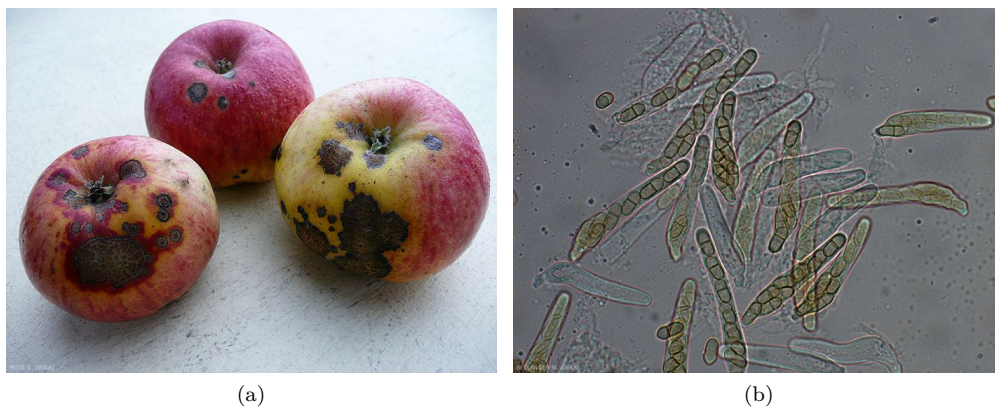


FIGURE 1 – (a) Pomme atteinte de la tavelure, (b) Asques de *Venturia Inaequalis*.

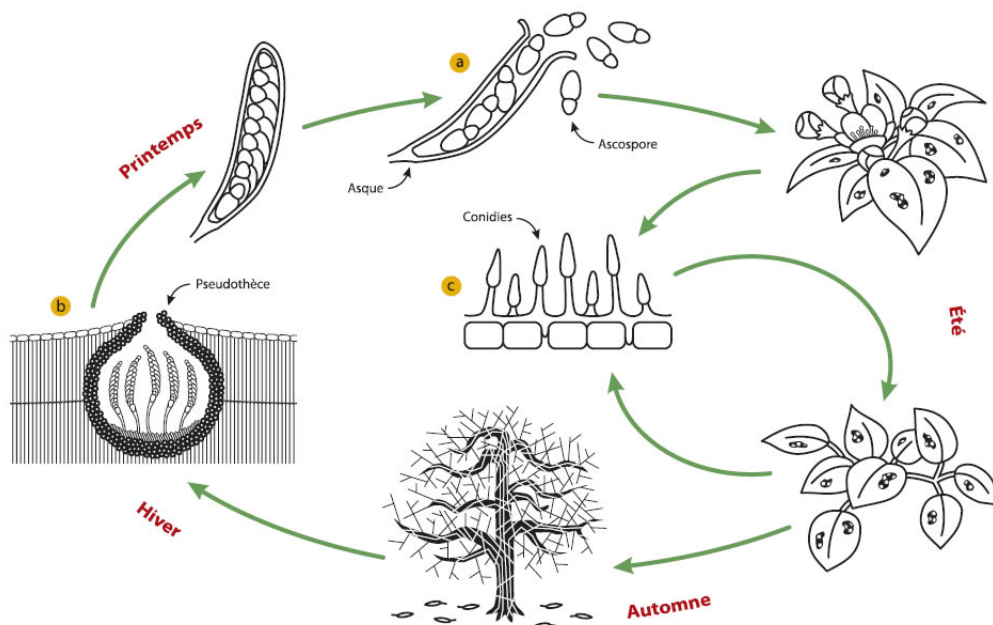


FIGURE 2 – Cycle biologique du champignon *Venturia Inaequalis*.

détruire le champignon. Très utiles, ils ralentissent considérablement le développement de la maladie et une utilisation répétée permet de préserver le fruit.

Néanmoins, les traitements représentent aussi une grande partie des dépenses nécessaires pour la protection des vergers de pommier et constituent un coût pour l'agriculteur. De plus, certains produits sont phytotoxiques et présentent un danger pour la santé des consommateurs. Ce risque est proportionnel à la fréquence d'application et aux doses utilisées. Or, cette fréquence d'application est trop élevée au vu de la réelle dynamique du champignon.

Être capable de limiter les traitements tout en préservant une bonne récolte constitue alors un enjeu majeur. L'idéal serait de prévoir à quels moments les champignons libèrent leurs spores afin de ne traiter les pommiers qu'aux moments adéquats. Un bon modèle permettant la prédiction des instants de libération de spores représente donc un grand intérêt agronomique.

3.3 Etat de l'art

La dynamique du champignon *Venturia Inaequalis* a fait l'objet de nombreuses études expérimentales en Europe [Jankowski and Masny, 2020]. Les informations recueillies pendant ces études ont permis d'établir un ensemble de règles de décisions déterministes qui visent à reproduire le cycle biologique du champignon. Elles sont à l'origine d'algorithmes qui simulent la dynamique du champignon, implémentés dans des logiciels RIMpro [Rancane et al., 2007]. Ces derniers simulent le risque d'infection sur les feuilles ou les fruits à partir des données climatiques et des données de libération de spores. Néanmoins les règles de décisions dépendent en grande partie des conditions expérimentales qui varient spatialement. De plus, elles ne tiennent pas compte d'aléa inhérent au processus de dispersion des spores.

3.4 L'objectif de stage

L'objectif de ce stage est de proposer un modèle stochastique de dispersion des spores afin de fournir un outil prédictif générique et d'ajuster ce modèle aux données expérimentales.

Compte tenu de la structure des données (temps d'arrivées des spores) le cadre de modélisation par des processus ponctuels s'est imposé. Dans ce stage j'ai étudié deux modèles de processus ponctuels : le processus de Poisson (homogène et non homogène) et le processus de Hawkes. Ce dernier, proposé à l'origine pour modéliser les données sismiques [Ogata, 1988] a également été utilisé en épidémiologie [Schoenberg et al., 2019].

Dans ce rapport on étudie tout d'abord les propriétés des modèles des processus ponctuels considérés, des algorithmes de simulations et les méthodes statistiques pour ces modèles. L'étude des modèles est suivie par l'application aux données expérimentales de la propagation de la tavelure.

Dans la section suivante, les notions de processus de comptage et de Poisson vont être définies. Des algorithmes de simulation et des procédures d'inférences statistiques vont ensuite être présentés. Cette section se base sur les polycopiés de [Decreusefond and Moyal, 2023], de [M.Deaconu, 2022] et de [Cadre, 2022].

4 Processus de comptage et de Poisson

4.1 Définition et propriété

Nous allons tout d'abord définir la notion de processus ponctuel et de comptage.

Définition 4.1. [Decreusefond and Moyal, 2023]

Un processus ponctuel intégrable est une suite strictement croissante de variables aléatoires positives (T_1, T_2, \dots) telle que $T_n \rightarrow \infty$ lorsque n tend vers l'infini presque-sûrement. Par convention, on adjoint à cette suite la variable aléatoire $T_0 = 0$ presque sûrement. Les (T_1, T_2, \dots) sont appelés les points du processus ponctuel, et représenteront les instants d'arrivées des événements. On peut tout aussi bien décrire la suite par l'écart de temps qui s'écoule entre deux arrivées successives : $W_{n+1} = T_{n+1} - T_n$ est la n -ième inter-arrivée. La donnée de la suite $(W_n, n \in \mathbb{N}^*)$ définit aussi un processus ponctuel par la relation $T_n = \sum_{i \leq n} W_i$. On notera finalement N_t , le nombre de points (d'arrivées), entre les instants 0 et t .

Définition 4.2. [Cadre, 2022]

Un processus stochastique $N = (N_t)_{t \geq 0}$ est un processus de comptage si p.s., ses trajectoires sont croissantes par saut d'amplitude 1, continues à droites et nulles à l'instant 0. Autrement dit, un processus de comptage N est caractérisé par les propriétés suivantes, avec probabilité 1 :

1. $N_0 = 0$:
2. N est croissante, continue à droite et constante entre deux instants de saut ;
3. $\Delta N_t \in \{0, 1\}$ pour tout $t \in \mathbb{R}^+$.

Remarque 4.1. Les définitions 4.1 et 4.2 sont équivalentes.

L'objet $(N_t)_{t \geq 0}$ désigne le processus de comptage, et N_t une variable aléatoire à valeur dans \mathbb{N} . La trajectoire d'un processus $(N_t)_{t \geq 0}$ est une fonction constante par morceaux et discontinue aux endroits d'instants d'arrivée. Sur la Figure 3, la trajectoire constante par morceaux du processus et les quatre instants d'arrivée (t_1, t_2, t_3, t_4) sont représentés.

On interprète les temps d'arrivées comme des occurrences d'événements. On considère par exemple un champignon qui libère ses spores dans le temps. Alors ces libérations peuvent être représentées par un processus ponctuel. Les T_i représentent les instants de libérations de spores, les W_i l'écart de temps s'écoulant entre deux instants et N_t le nombre de libérations de spores produits jusqu'à t .

Le processus de comptage le plus connu est le processus de Poisson, il est analysé dans la section 4.2.

4.2 Processus de Poisson homogène

Le processus de Poisson est utilisé dans de nombreux domaines comme l'informatique, les réseaux sociaux ou la biologie. Par exemple, la quantité d'appels à un centre téléphonique ou le nombre de personnes qui se présentent à une file d'attente peuvent se modéliser par un processus de Poisson.

Dans cette sous-section le processus de Poisson est défini de plusieurs manière et des outils de simulation et d'estimation de paramètres sont proposés.

Définition 4.3. [Decreusefond and Moyal, 2023]

Un **processus de Poisson homogène** $(N_t)_{t \geq 0}$ est un processus de comptage satisfaisant deux propriétés :

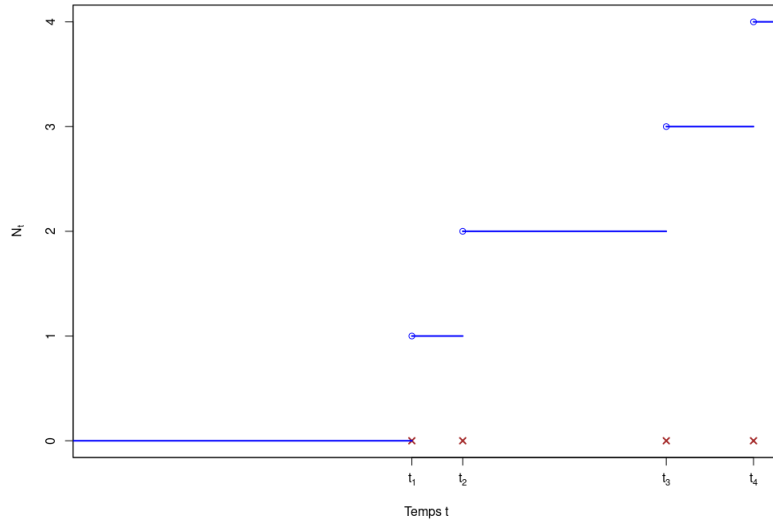


FIGURE 3 – Trajectoire d'un processus de comptage (en bleu) avec quatre instants d'arrivées (en marron).

1. Pour tout $p > 1$ et $0 < t_1 < t_2 < \dots < t_p$ les variables aléatoires $N_{t_1}, N_{t_2} - N_{t_1}, \dots, N_{t_p} - N_{t_{p-1}}$ sont indépendantes.
2. Pour tout t, s , la variable aléatoire $N_{t+s} - N_t$ suit une loi de Poisson de paramètre λs :

$$\mathbb{P}(N_{t+s} - N_t = k) = \exp(-\lambda s) \frac{(\lambda s)^k}{k!}.$$

Remarque 4.2. On se place sous les hypothèses de 4.3, alors :

$$\mathbb{P}(N_{t+s} - N_t = k) = \mathbb{P}(N_s = k).$$

On appelle cette propriété la propriété de stationnarité.

Le paramètre λ est appelée **l'intensité** du processus. On peut l'interpréter comme le nombre de points attendus dans un intervalle de temps de longueur 1.

Il est également possible de définir le processus de Poisson en tant qu'un processus de renouvellement.

Définition 4.4. Soient $\lambda > 0$ et $(W_i)_{i \in \mathbb{N}^*}$ des variables aléatoires i.i.d. suivant la loi exponentielle de paramètre $\lambda > 0$: pour tout $i, W_i \sim \mathcal{E}(\lambda)$. On note $T_n = \sum_{i=1}^n W_i$, alors le processus $(N_t)_{t \geq 0}$ défini par :

$$N_t = \sum_{i=1}^{+\infty} \mathbb{1}_{\{T_i \leq t\}}$$

est un **processus de Poisson homogène** de paramètre λ .

Une autre définition du processus de Poisson utilise une propriété d'uniforme distribution des instants d'arrivées.

Définition 4.5. [Decreusefond and Moyal, 2023]

Le processus ponctuel N est un processus de Poisson d'intensité λ si et seulement si les deux conditions suivantes sont réunies :

1. Pour tout $t \geq 0$ la variable aléatoire N_t suit une loi de Poisson de paramètre λt .
2. Conditionnellement à $\{N_t = n\}$, la famille (t_1, \dots, t_n) est uniformément distribuée sur $[0, t]$.

Proposition 4.1. Les définitions 4.3, 4.4 et 4.5 son équivalentes.

Démonstration. On commence par montrer $4.3 \Rightarrow 4.4$. Cette preuve se base sur les résultats de [Ruwet, 2007] page 26.

On considère $(N_t)_{t \in \mathbb{R}^+}$ un processus de Poisson défini en 4.3 et $(T_n)_{n \geq 1}$ la suite de ses instants d'arrivées. Soient $(s_i)_{1 \leq i \leq n}$ et $(t_i)_{1 \leq i \leq n}$ des réels strictement ordonnés. On considère l'évènement A défini par

$$A = \{0 < s_1 < T_1 < t_1 < s_2 < T_2 < t_2 < s_3 < \dots < t_{n-1} < s_n < T_n < t_n\}.$$

Nous cherchons à obtenir la fonction de densité jointe des temps T_i . En utilisant la propriété d'indépendance des accroissement et de stationnarité des accroissement il vient :

$$\begin{aligned} \mathbb{P}(A) &= \mathbb{P}(N_{s_1} = 0, N_{t_1} - N_{s_1} = 1, N_{s_2} - N_{t_1} = 0, \dots, N_{s_n} - N_{t_{n-1}} = 0 \\ &\quad, N_{t_n} - N_{s_n} \geq 1) \\ &= \mathbb{P}(N_{s_1} = 0) \mathbb{P}(N_{t_1} - N_{s_1} = 1) \mathbb{P}(N_{s_2} - N_{t_1} = 0) \dots \mathbb{P}(N_{s_n} - N_{t_{n-1}} = 0) \\ &\quad \mathbb{P}(N_{t_n} - N_{s_n} \geq 1) \\ &= e^{-\lambda s_1} \lambda (t_1 - s_1) e^{-\lambda(t_1 - s_1)} e^{-\lambda(s_2 - t_1)} \dots e^{-\lambda(s_n - t_{n-1})} (1 - e^{-\lambda(t_n - s_n)}) \\ &= \lambda^{n-1} (e^{-\lambda s_n} - e^{-\lambda t_n}) \prod_{i=1}^{n-1} (t_i - s_i) \\ &= \int_{s_n}^{t_n} \lambda e^{-\lambda y_n} dy_n \prod_{i=1}^{n-1} (t_i - s_i) \\ &= \int_{s_1}^{t_1} \int_{s_2}^{t_2} \dots \int_{s_n}^{t_n} e^{-\lambda y_n} \lambda^n dy_1 dy_2 \dots dy_n. \end{aligned}$$

La fonction de densité jointe des instants d'arrivées T_i est donc :

$$f_{T_1, \dots, T_n}(y_1, \dots, y_n) = e^{-\lambda y_n} \lambda^n.$$

On effectue un changement de variable pour retrouver les temps d'inter-arrivées. On pose $W_1 = T_1$, $W_1 + W_2 = T_2, \dots, W_1 + W_2 + \dots + W_n = T_n$, le jacobien de ce changement de variable est égal

à 1. La fonction de densité des temps d'inter-arrivées s'écrit :

$$f_{W_1, W_2, \dots, W_n}(w_1, w_2, \dots, w_n) = \lambda^n \exp(-\lambda \sum_{i=1}^n w_i). \quad (1)$$

On remarque que l'expression de la densité de la première arrivée s'écrit :

$$f_{W_1}(w_1) = \lambda \exp(-\lambda w_1).$$

Cette densité de probabilité est la densité d'une loi exponentielle de paramètre λ : $W_1 \sim \mathcal{E}(\lambda)$.

On cherche ensuite à connaître la densité de W_2 en sachant l'événement W_1 réalisé. Cela se note :

$$\begin{aligned} f_{W_2|W_1=w_1}(w_2) &= \frac{f_{(W_1, W_2)}(w_1, w_2)}{f_{W_1}(w_1)} \\ &= \frac{\lambda^2 e^{-\lambda w_1} e^{-\lambda w_2}}{\lambda e^{-\lambda w_1}} \\ &= \lambda e^{-\lambda w_2}. \end{aligned}$$

Il n'y a pas de terme en w_1 , ainsi les deux variables W_1 et W_2 sont indépendantes et W_2 suit une loi exponentielle de paramètre λ .

Pour le cas général, il suffit de considérer l'événement "tout les W_j on été réalisés" avec $1 \leq j \leq n-1$:

$$\begin{aligned} f_{W_j|W_{j-1}=w_{n-1}, \dots, W_1=w_1}(w_j) &= \frac{f_{W_1, \dots, W_j}(w_1, \dots, w_j)}{f_{W_1, \dots, W_{j-1}}(w_1, \dots, w_{j-1})} \\ &= \lambda e^{-\lambda w_j}. \end{aligned}$$

L'expression trouvée est indépendante de tout les anciens événements, on en déduit que

$$f_{W_j}(w_j) = \lambda e^{-\lambda w_j}.$$

Ce qui achève de montrer que toutes les variables $(W_j)_{1 \leq j \leq n}$ sont indépendantes et suivent une loi exponentielle de paramètre λ .

□

Démonstration. On s'intéresse à 4.4 \Rightarrow 4.5. La démonstration est basée sur les résultats de [Decreusefond and Moyal, 2023] page 16.

Soient $(W_i)_{i \in \mathbb{N}^*}$ des variables aléatoires indépendantes identiquement distribués de loi exponentielle de paramètre $\lambda > 0$ (i.e pour tout i $W_i \sim \mathcal{E}(\lambda)$).

On montre dans un premier temps que la variable aléatoire N_t suit une loi de Poisson de paramètre λt .

Soit $t > 0$ tel que $N_t = n$, alors en terme d'événements on a :

$$\{N_t = n\} = \{N_t \geq n\} \setminus \{N_t \geq n+1\} = \{T_n \leq t\} \setminus \{T_{n+1} \leq t\}.$$

On passe à la probabilité pour obtenir :

$$\begin{aligned}\mathbb{P}(N_t = n) &= \mathbb{P}(T_n \leq t) - \mathbb{P}[(T_{n+1} \leq t) \cap (T_n \leq t)] \\ &= \mathbb{P}(T_n \leq t) - \mathbb{P}(T_{n+1} \leq t).\end{aligned}$$

La loi de la somme de n variables aléatoires i.i.d. exponentielles de paramètres $\lambda > 0$ est la loi Gamma de paramètres $(n, \lambda) : \Gamma(n, \lambda)$. Ainsi la variable aléatoire T_n suit cette loi Gamma et on a :

$$\begin{aligned}\mathbb{P}(N_t = n) &= \int_0^t \frac{e^{-\lambda s} \lambda^n s^{n-1}}{(n-1)!} ds - \int_0^t \frac{e^{-\lambda s} \lambda^{n+1} s^n}{n!} ds \\ &= \int_0^t \frac{e^{-\lambda s} \lambda^n s^{n-1}}{(n-1)!} ds + \left[\frac{(\lambda s)^n e^{-\lambda s}}{n!} \right]_0^t - \int_0^t \frac{e^{-\lambda s} \lambda^n s^{n-1}}{(n-1)!} ds \\ &= \frac{(\lambda t)^n e^{-\lambda t}}{n!}.\end{aligned}$$

On en déduit que la variable aléatoire N_t suit une loi de Poisson de paramètre λt , i.e $N_t \sim \mathcal{P}(\lambda t)$.

Soient $T > 0, \lambda > 0$, $(T_i)_{1 \leq i \leq n+1}$ les variables aléatoires des instants des $(n+1)$ premières arrivées et $(W_i)_{1 \leq i \leq n+1}$ les variables aléatoires des temps d'inter-arrivées. On va montrer que conditionnellement à $\{N_T = n\}$ la famille (T_1, \dots, T_n) est uniformément répartie sur l'intervalle $(0, T]$.

Soient $[s_1, t_1], \dots, [s_n, t_n]$ des intervalles de temps disjoints et ordonnés tel que $t_n < T$.

On s'intéresse à l'événement :

$$\begin{aligned}B &= \{T_1 \in [s_1, t_1], \dots, T_n \in [s_n, t_n], N_T = n\} \\ &= \{T_1 \in [s_1, t_1], \dots, T_n \in [s_n, t_n], T_{n+1} > T\} \\ &= \{W_1 \in [s_1, t_1], \dots, \sum_{i=1}^n W_i \in [s_n, t_n], \sum_{i=1}^{n+1} W_i > T\}.\end{aligned}$$

Pour calculer la probabilité $\mathbb{P}(B)$, on utilise l'indépendance des variables W_i . On désigne par $f_{W_i}(w_i)$

la fonction de densité de la i ème inter-arrivée, ainsi :

$$\begin{aligned}
\mathbb{P}(B) &= \int \cdots \int \mathbb{I}_{s_1 < w_1 < t_1} \cdots \mathbb{I}_{s_n < \sum_{i=1}^{n-1} w_i < t_n} \mathbb{I}_{T < \sum_{i=1}^n w_i} f_{W_1}(w_1) \cdots f_{W_n}(w_n) f_{W_{n+1}}(w_{n+1}) dw_1 \cdots dw_{n+1} \\
&= \int_{s_1}^{t_1} \lambda e^{-\lambda w_1} \left(\cdots \left(\int_{s_n - \sum_{i < n} w_i}^{t_n - \sum_{i < n} w_i} \lambda e^{-\lambda w_n} \left(\int_{T - \sum_{i \leq n} w_i}^{+\infty} \lambda e^{-\lambda w_{n+1}} dw_{n+1} \right) dw_n \right) \cdots \right) dw_1 \\
&= \int_{s_1}^{t_1} \lambda e^{-\lambda w_1} \left(\cdots \left(\int_{s_n - \sum_{i < n} w_i}^{t_n - \sum_{i < n} w_i} \lambda e^{-\lambda w_n} e^{-\lambda(T - \sum_{i \leq n} w_i)} \right) \cdots \right) dw_1 \\
&= \lambda^n e^{-\lambda T} \int_{s_1}^{t_1} \left(\cdots \left(\int_{s_n - \sum_{i < n} w_i}^{t_n - \sum_{i < n} w_i} dw_n \right) \cdots \right) dw_1 \\
&= \lambda^n e^{-\lambda T} \prod_{i=1}^n (t_i - s_i).
\end{aligned}$$

On divise par $\mathbb{P}(N_T = n)$ afin de retrouver la probabilité conditionnelle recherchée. La variable aléatoire N_t suit une loi de Poisson de paramètre λt , il vient alors :

$$\begin{aligned}
\mathbb{P}\{T_1 \in [s_1, t_1], \dots, T_n \in [s_n, t_n] | N_T = n\} &= \frac{\mathbb{P}\{T_1 \in [s_1, t_1], \dots, T_n \in [s_n, t_n], T_{n+1} > T\}}{\mathbb{P}\{N_T = n\}} \\
&= \frac{\lambda^n e^{-\lambda T} \prod_{i=1}^n (t_i - s_i)}{\frac{(\lambda T)^n e^{-\lambda T}}{n!}} \\
&= n! \prod_{i=1}^n \frac{(t_i - s_i)}{T}.
\end{aligned}$$

On vient de trouver la distribution du vecteur (T_1, \dots, T_n) conditionnellement à $\{N_T = n\}$.

On considère maintenant $(U_i)_{1 \leq i \leq n}$ des variables aléatoires i.i.d. suivant une loi uniforme sur $[0, T]$. Alors on a :

$$\mathbb{P}\{U_1 \in [s_1, t_1], \dots, U_n \in [s_n, t_n]\} = \prod_{i=1}^n \frac{(t_i - s_i)}{T}.$$

En notant par $U_{1:1}, \dots, U_{1:n}$ le réordonnement croissant des variables aléatoires U_i il vient :

$$\mathbb{P}\{U_{1:1} \in [s_1, t_1], \dots, U_{1:n} \in [s_n, t_n]\} = n! \prod_{i=1}^n \frac{(t_i - s_i)}{T}.$$

Donc conditionnellement à $\{N_T = n\}$, le vecteur (T_1, \dots, T_n) a la même distribution que le vecteur de statistiques d'ordre $(U_{1:1}, \dots, U_{1:n})$. \square

Démonstration. On s'intéresse à la dernière implication 4.5 \Rightarrow 4.3. La démonstration se base sur le polycopié de [Decreusefond and Moyal, 2023] page 16 – 17.

On s'intéresse à la propriété d'indépendance de la Définition 4.3.

On s'intéresse tout d'abord à l'indépendance des accroissements.

Soient $t_0 < t_1 < \dots < t_n$ des réels positifs et $i_0 < i_1 < \dots < i_{n-1}$ des entiers. On considère la probabilité d'avoir exactement i_l points dans le l ème intervalle pour tout $0 \leq l \leq n-1$. On veut montrer que :

$$\mathbb{P}\left(\bigcap_{l=0}^{n-1} \{N_{t_{l+1}} - N_{t_l} = i_l\}\right) = \prod_{l=0}^{n-1} \mathbb{P}\{N_{t_{l+1}} - N_{t_l} = i_l\}.$$

On remarque que l'on peut toujours écrire

$$\mathbb{P}\left(\bigcap_{l=0}^{n-1} \{N_{t_{l+1}} - N_{t_l} = i_l\}\right) = \sum_{k \in \mathbb{N}} \mathbb{P}\left(\bigcap_{l=0}^{n-1} \{N_{t_{l+1}} - N_{t_l} = i_l | N_{t_n} = k\}\right) \mathbb{P}(N_{t_n} = k). \quad (2)$$

Cette somme est non nulle seulement pour $k_0 = \sum l = 0^{n-1} i_l$. Or on sait que les arrivées sont uniformément réparties sur $[0, t_n]$. On considère alors la probabilité que k_0 points soient uniformément répartis sur $[0, t_n]$ avec i_1 points sur $[t_0, t_1]$, i_2 dans $[t_1, t_2]$ etc. Chaque point se trouve dans un intervalle $[t_{i-1}, t_i]$ avec probabilité $\frac{t_i - t_{i-1}}{t_n}$. On utilise la fonction multinomiale pour exprimer cette probabilité. Il y a i_l points dans le l ème intervalle avec pour probabilité la longueur de l'intervalle. Ce qui s'écrit :

$$\mathbb{P}\left(\bigcap_{l=0}^{n-1} \{N_{t_{l+1}} - N_{t_l} = i_l | N_{t_n} = k_0\}\right) = \frac{k_0!}{i_1! i_2! \dots i_n!} \prod_{l=0}^{n-1} \left(\frac{t_{l+1} - t_l}{t_n}\right)^{i_l}.$$

La probabilité 2 devient alors :

$$\begin{aligned} \mathbb{P}\left(\bigcap_{i=0}^{n-1} \{N_{t_{i+1}} - N_{t_i} = i_i\}\right) &= \mathbb{P}(N_{t_n} = k_0) \frac{k_0!}{i_1! i_2! \dots i_n!} \prod_{l=0}^{n-1} \left(\frac{t_{l+1} - t_l}{t_n}\right)^{i_l} \\ &= \frac{e^{-\lambda t_n} (\lambda t_n)^{k_0}}{k_0!} \frac{k_0!}{i_1! i_2! \dots i_n!} \prod_{l=0}^{n-1} \left(\frac{t_{l+1} - t_l}{t_n}\right)^{i_l} \\ &= \prod_{l=0}^{n-1} \left(e^{-\lambda(t_{l+1} - t_l)} \frac{(\lambda(t_{l+1} - t_l))^{i_l}}{i_l!}\right) \\ &= \prod_{l=0}^{n-1} \mathbb{P}(N_{t_{l+1}} - N_{t_l} = i_l). \end{aligned}$$

Ainsi les variables aléatoires $(N_{l+1} - N_l)_{0 \leq l < n}$ sont indépendantes car la probabilité de l'intersection des événements correspond au produit des probabilités. La propriété de l'indépendance des accroissements est donc satisfaite.

On s'intéresse maintenant à la loi de $N_{t+s} - N_t$. On utilise l'équation précédente avec les valeurs suivantes : $n = 2, t_1 = t, t_2 = t+s, i_0 = i$ et $i_1 = j$. On obtient par indépendance des accroissements :

$$\mathbb{P}(N_t = i, N_{t+s} - N_t = j) = \mathbb{P}(N_t = i) \mathbb{P}(N_{t+s} - N_t = j)$$

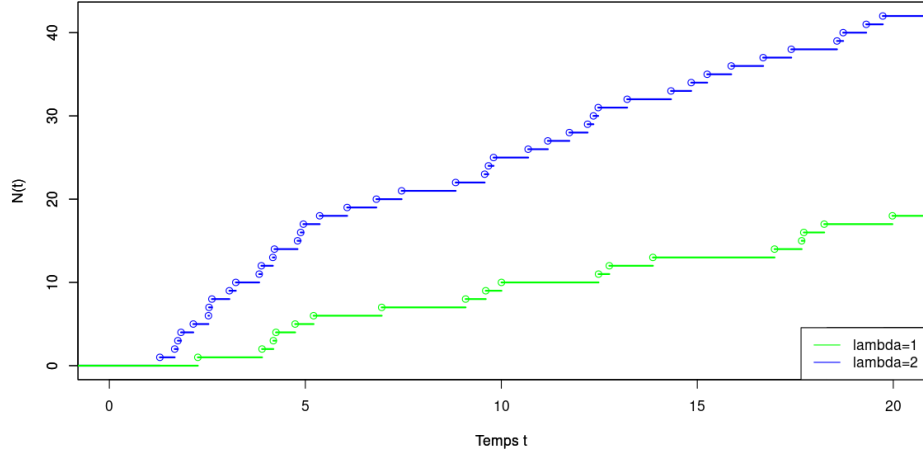


FIGURE 4 – Trajectoires de processus de Poisson homogène pour $\lambda = 1$ et $\lambda = 2$

On somme sur toutes les valeurs de i pour obtenir :

$$\mathbb{P}(N_{t+s} - N_t = j) = \mathbb{P}(N_s = j).$$

□

Avec ces définitions, le processus de Poisson homogène peut être vu de plusieurs façons.

- Un processus qui a des accroissements indépendants et stationnaire et dont le nombre de point dans un intervalle de temps suit une loi de Poisson.
- Un processus de renouvellement où les temps d'inter-arrivées suivent une loi exponentielle.
- Un processus dont le nombre d'instants d'arrivée dans un intervalle donné suit une loi de Poisson et où ces points sont uniformément distribués conditionnellement à cette loi de Poisson.

C'est souvent cette dernière caractérisation que est retenue. La Figure 4 montre deux trajectoires de processus de Poisson homogène avec $\lambda = 1$ et $\lambda = 2$.

Une autre propriété des processus de Poisson concerne les accroissements.

Propriété 4.1. Soit $(N_t)_{t \geq 0}$ un processus de Poisson homogène d'intensité $\lambda > 0$. Alors pour $t \geq 0$ et $h > 0$ suffisamment petit on a les trois propriétés suivantes :

$$\mathbb{P}(N(t, t+h] = 0) = 1 - \lambda h + o(h) \quad (3)$$

$$\mathbb{P}(N(t, t+h] = 1) = \lambda h + o(h) \quad (4)$$

$$\mathbb{P}(N(t, t+h] \geq 2) = o(h). \quad (5)$$

Démonstration. Soit h suffisamment petit, alors d'après l'hypothèse de stationnarité du processus de Poisson 4.4 on a :

$$\mathbb{P}(N(t, t+h] = k) = \mathbb{P}(N(0, h] = k)$$

pour tout $k \in \mathbb{N}$.

En utilisant un développement limité il vient :

$$\begin{aligned}\mathbb{P}(N(t, t+h] = 0) &= 1 - \lambda h + o(h). \\ \mathbb{P}(N(t, t+h] = 1) &= \lambda h + o(h). \\ \mathbb{P}(N(t, t+h] \geq 2) &= 1 - \mathbb{P}(N(t, t+h] = 0) - \mathbb{P}(N(t, t+h] = 1) = o(h).\end{aligned}$$

□

4.2.1 Simulation

On souhaite simuler un processus de Poisson homogène de paramètre λ sur l'intervalle de temps $(0, T]$.

La première manière de simuler un processus de Poisson repose sur la définition 4.5. On simule d'abord le nombre de points N du processus selon la loi de Poisson $\mathcal{P}(\lambda T)$. Ensuite on simule un échantillon de N points répartis uniformément sur $(0, T]$ (voir l'algorithme 1).

Le code en langage **R** est donné en annexe 8.1.

Algorithm 1 Simulation du processus de Poisson par la propriété uniforme. [Decreusefond and Moyal, 2023]

Require: Intensité λ , limite de temps T

$Liste \leftarrow [], t \leftarrow 0$

$N \leftarrow \mathcal{P}(\lambda T)$ Simule une loi de Poisson de paramètre λT .

for $1 \leq i \leq N$ **do**

$U \leftarrow Unif([0, T])$ réalisation d'une loi uniforme sur l'intervalle de temps $(0, T]$.

$Liste \leftarrow [Liste, U]$

end for

$Liste \leftarrow \text{Tri croissant de } Liste$

return Liste

La seconde approche se base sur la Définition 4.4. Elle consiste à tirer un échantillon de lois exponentielles indépendantes de paramètre $\lambda > 0$ et de faire la somme. L'algorithme 2 présente le code pour simuler une trajectoire d'un processus de Poisson homogène. Le code écrit en **R** se trouve en annexe 8.2.

4.2.2 Estimation des paramètres

Pour estimer la valeur du paramètre λ à partir de données, on s'intéresse à la fonction de vraisemblance du processus. On rappelle que la fonction de vraisemblance est définie par :

Algorithm 2 Simulation du processus de Poisson par renouvellement.

Require: Intensité λ , Limite de temps T

$Liste \leftarrow [], t \leftarrow 0$

while $t < T$ **do**

$W \leftarrow \mathcal{E}(M)$ réalisation d'une loi exponentielle de paramètre λ .

$t \leftarrow t + W$

$Liste \leftarrow [Liste, t]$

end while

Supprimer le dernier élément de $Liste$.

return $Liste$

Proposition 4.2. Soit (t_1, \dots, t_k) une réalisation d'un processus de Poisson homogène sur un intervalle de temps $(0, T]$. Alors la fonction de vraisemblance du paramètre λ est donnée par :

$$L(\lambda|t_1, \dots, t_k, T) = \lambda^k \exp(-\lambda T). \quad (6)$$

Démonstration. Cette preuve se base sur les résultats de [Daley and Vere-Jones, 2003]. Soit $T > 0$ et $\Delta \ll 1$, on suppose qu'il y a k observations sur $(0, T]$. On veut calculer la probabilité jointe que les k points appartiennent aux intervalles $(t_i - \Delta, t_i]$ avec $1 \leq i \leq k$,

$$N(t_1 - \Delta) = 0; N(t_1 - \Delta, t_1] = 1, \dots, N(t_i - \Delta) = 0; N(t_i - \Delta, t_i] = 1, \dots, N(t_k, T] = 0. \quad (7)$$

En utilisant les propriétés de la définition 4.3, il vient :

$$\begin{aligned} \mathbb{P}(7) &= e^{-\lambda T} \prod_{j=1}^k \lambda \Delta \\ &= e^{-\lambda T} (\lambda \Delta)^k. \end{aligned}$$

La densité jointe est égale à :

$$\begin{aligned} L(\lambda|t_1, \dots, t_k, T) &= \lim_{\Delta \rightarrow 0} \frac{\mathbb{P}(7)}{\Delta^k} \\ &= \lambda^k e^{-\lambda T}. \end{aligned}$$

□

Par conséquent la log-vraisemblance de (6) est égale à :

$$l(\lambda|t_1, \dots, t_k, T) = k \ln(\lambda) - \lambda T. \quad (8)$$

Cette fonction est convexe et atteint un maximum au point $\hat{\lambda} = \frac{k}{T}$.

Afin de tester l'hypothèse selon laquelle le processus ponctuel est un processus de Poisson d'intensité λ , il suffit de tester si les points sont uniformément distribués (voir 4.5) ou si les temps d'inter-arrivées sont i.i.d. et suivent une loi exponentielle de paramètre $\hat{\lambda}$ (voir 4.4). Cela peut être fait avec un test de Kolmogorov-Smirnov.

4.3 Processus de Poisson non homogène

En général, l'hypothèse d'une intensité constante est trop restrictive. La plupart du temps le flux d'arrivée dans une file d'attente n'est pas constant dans le temps. Il y a des moments où il y a plus de personnes qui arrivent dans la file et d'autres moments moins chargés. En reprenant l'exemple de la dynamique du champignon *Venturia Inaequalis*, la libération de spores est influencée par la chaleur ou l'humidité qui dépendent du temps. Cela se traduit par une variation de l'intensité au cours du temps. C'est la raison pour laquelle on introduit la notion de processus de Poisson non homogène.

Définition 4.6. [Cadre, 2022] Un **processus de Poisson non homogène** $(N_t)_{t \geq 0}$ est un processus de comptage satisfaisant deux propriétés :

1. Pour tout $p > 1$ et $0 < t_1 < t_2 < \dots < t_p$: les variables aléatoires $N_{t_1}, N_{t_2} - N_{t_1}, \dots, N_{t_p} - N_{t_{p-1}}$ sont indépendantes.
2. Il existe une fonction $\lambda(t)$, positive et intégrable tel que pour tout $0 \leq s \leq t$: le nombre de points sur l'intervalle de temps $[s, t]$ suit une loi de Poisson de paramètre $\int_0^t \lambda(u) du$:

$$N_t - N_s \sim \mathcal{P}\left(\int_s^t \lambda(u) du\right). \quad (9)$$

Exemple 4.1. [Baddeley et al., 2016]

L'intensité log-linéaire permet d'inclure des covariables dans le modèle.

Une intensité d'un processus de Poisson non homogène est log-linéaire lorsqu'elle peut s'écrire sous la forme :

$$\lambda_\theta(t) = \exp(B(t) + \theta^\top Z(t)). \quad (10)$$

Où $B(t)$ est une fonction connue, $\theta = (\theta_1, \dots, \theta_p)$ est le vecteur des paramètres et $Z(t) = (Z_1(t), \dots, Z_p(t))$ le vecteur des covariables.

Définition 4.7. On appelle compensateur l'intégrale de la fonction d'intensité, il est noté $\Lambda(\cdot)$.

$$\Lambda(t) = \int_0^t \lambda(s) ds. \quad (11)$$

Remarque 4.3. L'espérance d'une loi de Poisson est égale à son paramètre, en utilisant l'équation (9), il vient :

$$\mathbb{E}[N_t - N_s] = \int_s^t \lambda(u) du = \Lambda(t) - \Lambda(s). \quad (12)$$

Propriété 4.2. Soit $(N_t)_{t \geq 0}$ un processus de Poisson non homogène, $\Lambda(\cdot)$ son compensateur et $\mathcal{H}(t)$ une filtration qui désigne l'historique du processus : $\mathcal{H}(t) = \{N_s, s \leq t\}$ avec t_i .

On considère $M(t) = N_t - \Lambda(t)$, alors $M(t)$ est une martingale.

Démonstration. La variable aléatoire $M(t)$ est intégrable et $\mathcal{H}(t)$ -adaptée. On vérifie ensuite pour tout $0 \leq t \leq s$ que :

$$\mathbb{E}[M(s)|\mathcal{H}(t)] = M(t).$$

$$\mathbb{E}[M(s)|\mathcal{H}(t)] = \mathbb{E}[N_s - \Lambda(s)|\mathcal{H}(t)]$$

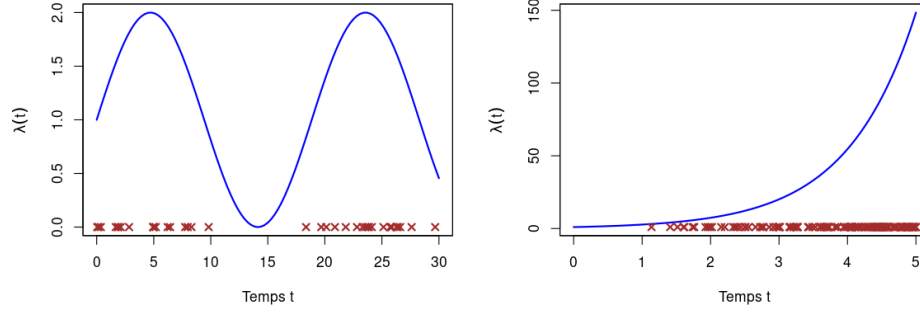


FIGURE 5 – Simulation de deux processus de Poisson avec les instants d’arrivées (marron) et la courbe d’intensité (bleu). a) $\lambda(t) = 1 + \sin(t/3)$, b) $\lambda(t) = \exp(t)$.

L’espérance est linéaire et $\Lambda(\cdot)$ est une fonction déterministe.

$$\begin{aligned}
 \mathbb{E}[M(s)|\mathcal{H}(t)] &= \mathbb{E}[N_s|\mathcal{H}(t)] - \Lambda(s) \\
 &= \mathbb{E}[N_t + N(t, s)|\mathcal{H}(t)] - \Lambda(s) \\
 &= \mathbb{E}[N_t|\mathcal{H}(t)] + \mathbb{E}[N(t, s)|\mathcal{H}(t)] - \Lambda(s) \\
 &= N_t + \mathbb{E}[N(t, s)] - \Lambda(s) \\
 &= N_t + \Lambda(s) - \Lambda(t) - \Lambda(s) \\
 &= M(t)
 \end{aligned}$$

en notant $N(t, s] = N_s - N_t$.

□

La Figure 5 présente deux trajectoires de processus de Poisson non homogène.

4.3.1 Simulation

On va présenter deux algorithmes de simulation du processus de Poisson non homogène : la procédure de l’inverse du compensateur et l’algorithme d’amincissement de [Lewis and Shedler, 1979]. La première se base sur la propriété 4.3.

Propriété 4.3. Soit $\lambda(\cdot)$ une intensité strictement positive tel que $\lim_{t \rightarrow +\infty} \Lambda(t) = +\infty$. On considère $(s_i)_{i \in \mathbb{N}}$ un processus de Poisson d’intensité unitaire sur \mathbb{R} et $t_i = \Lambda^{-1}(s_i)$, alors $(t_i)_{i \in \mathbb{N}}$ forment un processus de Poisson non homogène d’intensité $\lambda(\cdot)$.

Démonstration. On utilise le raisonnement de [Rasmussen, 2011].

On suppose que l'on dispose de k points (s_1, \dots, s_k) d'un processus de Poisson homogène d'intensité 1. Soient (t_1, \dots, t_k) les points d'un processus de Poisson non homogène tel que $t_i = \Lambda(s_i)$ pour tout $1 \leq i \leq k$.

On considère les prochains points s_{k+1} et $t_{k+1} = \Lambda^{-1}(s_{k+1})$. Le point s_{k+1} est une réalisation de la variable aléatoire S_{k+1} tel que $S_{k+1} = s_k + W_{k+1}$ avec W_{k+1} la variable aléatoire de la $(k+1)$ ème inter-arrivée du processus de Poisson d'intensité 1.

On veut montrer que le point t_{k+1} appartient au processus de Poisson non homogène d'intensité $\lambda(\cdot)$.

On considère la fonction de répartition du $(k+1)$ ème point du processus de Poisson d'intensité $\lambda(\cdot)$ désignée par $F_{T_{k+1}}(t)$.

$$\begin{aligned} F_{T_{k+1}}(t) &= \mathbb{P}(T_{k+1} \leq t) \\ &= \mathbb{P}(\Lambda^{-1}(s_k + W_{k+1}) \leq t) \\ &= \mathbb{P}(s_k + W_{k+1} \leq \Lambda(t)) \\ &= \mathbb{P}(W_{k+1} \leq \Lambda(t) - s_k) \\ &= 1 - \exp(-(\Lambda(t) - \Lambda(t_k))) \text{ car } W_{k+1} \sim \mathcal{E}(1) \\ &= 1 - \exp\left(-\int_{t_k}^t \lambda(s) ds\right). \end{aligned}$$

Il reste à montrer que cette expression est la fonction de répartition de la $(k+1)$ ème arrivée du processus de Poisson d'intensité $\lambda(\cdot)$. Soit $\Delta << 1$, on considère la probabilité :

$$\begin{aligned} \mathbb{P}(T_{k+1} \in (t - \Delta, t]) &= \mathbb{P}(N(t_k, t - \Delta] = 0, N(t - \Delta, t] = 1) \\ &= \int_{t-\Delta}^t \lambda(s) ds \exp\left(-\int_{t_k}^t \lambda(s) ds\right). \end{aligned} \tag{13}$$

On laisse tendre Δ vers 0 pour obtenir la densité :

$$\begin{aligned} f_{T_{k+1}}(t) &= \lim_{\Delta \rightarrow 0} \frac{(13)}{\Delta} \\ &= \lambda(t) \exp\left(-\int_{t_k}^t \lambda(s) ds\right). \end{aligned}$$

On en déduit que la fonction de répartition du temps d'arrivée t_k s'écrit :

$$F_{T_{k+1}}(t) = 1 - \exp\left(-\int_{t_k}^t \lambda(s) ds\right).$$

Ce qui achève la démonstration. □

On distingue ensuite deux cas de figure. Soit l'expression de l'inverse du compensateur est connue, soit elle n'est pas connue.

Cas numéro 1 : l'inverse du compensateur est connue

Pour une expression de l'inverse du compensateur connue, le pseudo code est donné dans l'algorithme 3. Il présente des instructions pour simuler un processus de Poisson non homogène d'intensité $\lambda(\cdot)$ sur l'intervalle de temps $(0, T]$. Le code écrit en **R** est donné en annexe 8.3.

Algorithm 3 Simulation du processus de Poisson non homogène par l'inverse du compensateur 1

Require: Le compensateur $\Lambda(\cdot)$, l'inverse du compensateur $\Lambda^{-1}(\cdot)$, une borne de temps T .

```

1:  $T\_aux \leftarrow \Lambda(T)$ 
2:  $Liste \leftarrow$  simulation d'un processus de Poisson homogène d'intensité 1 sur  $(0, T\_aux]$ .
3:  $Longueur \leftarrow Taille(Liste)$  Le nombre d'éléments de  $Liste$ .
4: for  $i \in Longueur$  do
5:    $Liste[i] \leftarrow \Lambda^{-1}(Liste[i])$ 
6: end for
7: return  $Liste$ 
```

L'algorithme est effectué en deux étapes. On commence par rechercher l'intervalle de temps dans lequel on simule un processus de Poisson homogène. Puis, chaque point généré est transformé par l'inverse du compensateur.

Cas numéro 2 : l'inverse du compensateur n'est pas connue

L'expression de l'inverse du compensateur n'est pas toujours accessible analytiquement. Dans ce cas, un processus de Poisson non homogène peut être construit de manière itérative. On ne connaît pas l'inverse du compensateur mais on peut en trouver une approximation à chaque nouveau point.

On se place dans les hypothèse de la propriété 4.3. On considère $(s_i)_{1 \leq i \leq k}$ les points d'un processus de Poisson homogène d'intensité 1 et $(t_i)_{1 \leq i \leq k}$ leur transformée par le compensateur. On considère S_{k+1} la fonction de répartition du $(k+1)$ -ième point du processus de Poisson et T_{k+1} la fonction de répartition du $(k+1)$ -ième point du processus de Poisson d'intensité $\lambda(\cdot)$, on a alors :

$$\begin{aligned} S_{k+1} - s_k &\sim \mathcal{E}(1) \\ \Lambda(T_{k+1}) - \Lambda(t_k) &\sim \mathcal{E}(1). \end{aligned}$$

Or pour simuler une loi exponentielle, on peut générer une loi uniforme :

$$\Lambda(t_{k+1}) - \Lambda(t_k) = -\ln(U) \text{ avec } U \sim Unif(0, 1).$$

Donc il revient à trouver t_{k+1} tel que l'expression ci-dessus soit satisfaite. L'algorithme 4 donne le pseudo code pour simuler le processus de cette manière. Le code **R** est disponible en annexe 8.4.

Algorithm 4 Simulation du processus de Poisson non homogène par l'inverse du compensateur 2

Require: Le compensateur $\Lambda(\cdot)$, une borne de temps T

```
1:  $Liste \leftarrow []$ 
2:  $t_k \leftarrow 0$  le dernier point de  $Liste$ , 0 initialement
3: while  $t_k \leq T$  do
4:    $U \leftarrow Unif(0, 1)$  simulation d'une loi uniforme sur l'intervalle de temps  $[0, 1]$ .
5:   Trouver  $t$  tel que  $\Lambda(t) - \Lambda(t_k) + \ln(U) = 0$ 
6:    $t_k \leftarrow t$ 
7:   if Si  $t_k \leq T$  then
8:     Ajouter  $t_k$  à  $Liste$ 
9:   end if
10: end while
11: return  $Liste$ 
```

La deuxième approche pour simuler un processus de Poisson non homogène s'appelle la méthode **d'amincissement**. Cette procédure a été introduite par [Lewis and Shedler, 1979]. Une borne supérieure pour l'intensité est requise pour utiliser cette méthode.

Le principe consiste à simuler un processus de Poisson homogène d'intensité la borne supérieure sur l'intervalle de temps considéré. Puis pour chaque point obtenu on génère une loi uniforme U sur l'intervalle $(0, M)$. Si la valeur de U est inférieure à la valeur de l'intensité alors le point est accepté.

Le pseudo code pour simuler un processus de Poisson non homogène, d'intensité $\lambda(\cdot)$ bornée par une constante positive M sur un intervalle de temps $(0, T]$ est fourni par l'Algorithme 5. Le code est fourni en annexe 8.5.

Algorithm 5 Simulation du processus de Poisson non homogène par amincissement.

Require: Intensité $\lambda(\cdot)$, borne supérieure M , Temps final T

```
1:  $Liste \leftarrow []$ 
2:  $Liste\_aux \leftarrow$  simulation d'un processus de Poisson homogène d'intensité  $M$  sur  $(0, T]$ .
3:  $k \leftarrow$  Longueur( $Liste\_aux$ )
4:  $U \leftarrow (U_1, \dots, U_k)$  avec  $U_i \sim Unif(0, M)$ .
5: for  $i \in [1, k]$  do
6:   if  $U[i] \leq \lambda(Liste\_aux[i])$  then
7:      $Liste \leftarrow [Liste, Liste\_aux[i]]$ 
8:   end if
9: end for
10: return  $Liste$ 
```

La Figure 6 illustre la procédure d'amincissement.

4.3.2 Inférence

Estimation des paramètres Pour estimer les paramètres d'un Poisson non homogène, on s'intéresse à la fonction de vraisemblance définie comme suit :

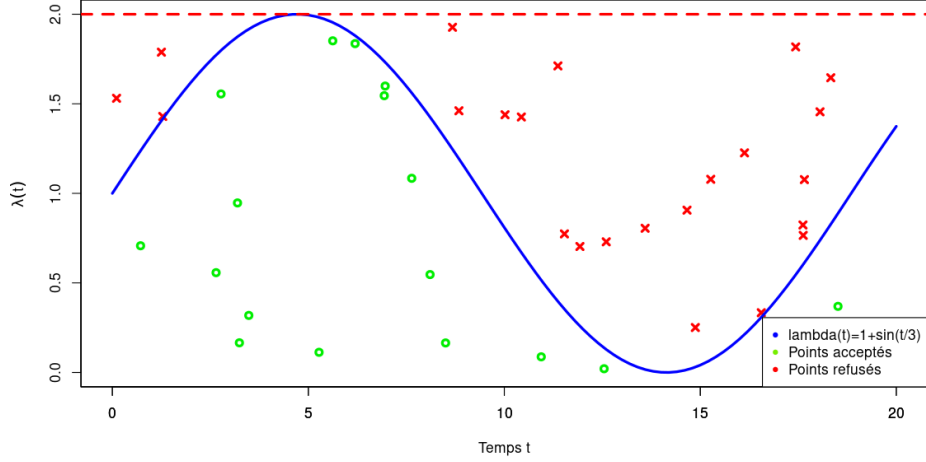


FIGURE 6 – Illustration de la méthode d’amincissement pour une intensité périodique $\lambda(t) = 1 + \sin(t/3)$ (tracée en bleue). La droite horizontale rouge $M = 2$ est sa borne supérieure, les cercles verts sous la courbe sont les points acceptés et les croix rouges au dessus de la courbe sont les points refusés). Les abscisses des cercles verts sont les instants d’arrivée du processus simulé.

Définition 4.8. Soit (t_1, \dots, t_n) une réalisation d’un processus de Poisson non homogène d’intensité $\lambda_\theta(\cdot)$ où $\theta \in \Theta$ (l’espace des paramètres) sur un intervalle de temps $(0, T]$. La fonction de vraisemblance du paramètre θ est donnée par :

$$L(\theta|t_1, \dots, t_k, T) = \prod_{i=1}^k \lambda_\theta(t_i) \exp\left(-\int_0^T \lambda_\theta(u) du\right). \quad (14)$$

Pour garantir l’existence et l’unicité du maximum de vraisemblance des conditions supplémentaires pour la fonction d’intensité sont nécessaire (voir [Cadre, 2022] page 91-93). Néanmoins dans le cas d’une intensité log-linéaire 4.1, la fonction de vraisemblance d’un processus de Poisson existe et est unique. Il est ainsi possible d’étudier le gradient du MLE, la Hessienne et les propriétés asymptotiques de l’estimateur. Ces résultats sont basés sur le livre de [Baddeley et al., 2016]

Définition 4.9. Pour un intensité log-linéaire définie en 4.1, la log-vraisemblance prend la forme suivante :

$$\log L(\theta|t_1, \dots, t_k, T) = \sum_{i=1}^k B(t_i) + \theta^\top \sum_{i=1}^k Z(t_i) - \int_0^T \exp(B(s) + \theta^\top Z(s)) ds. \quad (15)$$

où T désigne la borne de temps, t_i le i -ème instant d’arrivée et $k = n(T)$ le nombre de points observés sur la période de temps $(0, T]$.

Définition 4.10. *Le maximum de vraisemblance (MLE) est solution de l'équation de score $U(\theta) = 0$ où la fonction de score est :*

$$U(\theta) = \sum_{i=1}^k Z(t_i) - \int_0^T Z(s) \lambda_\theta(s) ds. \quad (16)$$

Définition 4.11. *La matrice d'information de Fisher est une matrice de taille $(p \times p)$ définie par :*

$$(I_\theta)_{i,j} = \int_0^T Z_i(s) Z_j(s) \lambda_\theta(s) ds. \quad (17)$$

Pour le modèle *log*–linéaire le MLE $\hat{\theta}$ est consistant, asymptotiquement normal et asymptotiquement efficace dans un large domaine [Rathbun and Cressie, 1994].

Validation du modèle. La validation du modèle se base sur la propriété 4.4.

Propriété 4.4. *On suppose que $\lambda(\cdot)$ est une intensité d'un processus de Poisson non homogène tel que pour tout $t \geq 0, \lambda(t) > 0$ et $\lim_{t \rightarrow +\infty} \Lambda(t) = +\infty$.*

On considère $(t_i)_{i \in \mathbb{N}}$ un processus de Poisson d'intensité $\lambda(\cdot)$ sur \mathbb{R} et $s_i = \Lambda(t_i)$, alors $(s_i)_{i \in \mathbb{N}}$ forment un processus de Poisson homogène d'intensité 1.

Cette propriété peut être démontrée de la même façon que la propriété 4.3.

La vérification du modèle consiste à vérifier que les temps d'inter-arrivées

$$R(i) = \Lambda(t_i) - \Lambda(t_{i-1})$$

appelés résidus suivent la loi exponentielle de paramètre 1. Des diagnostics graphiques comme un histogramme ou un test d'adéquation comme le test de Kolmogorov-Smirnov permettent de le faire.

5 Processus de Hawkes

La propriété centrale des processus de Poisson est l'indépendance des accroissements. Or il existe des processus dont les événements sont inter-dépendants. Pour caractériser les processus de ce type, on introduit la notion d'intensité conditionnelle.

Dans cette section on suivra essentiellement l'article de [Rasmussen, 2011] et l'article de [Laub et al., 2015].

5.1 Intensité conditionnelle

Un processus ponctuel peut être caractérisé par la loi jointe des instants d'arrivées. On considère un processus ponctuel $T = (T_i)_{i \geq 1}$. On désigne par $\mathcal{H}(t)$ l'historique du processus. Plus précisément $\mathcal{H}(\cdot)$ est une suite croissante de tribus pour l'inclusion. Dans notre cas, elle désigne l'ensemble des instants d'arrivées qui se sont produits jusqu'à l'instant t : $\mathcal{H}(t) = \{N_s : s \leq t\}$.

La fonction de répartition de la $(k + 1)$ -ième arrivée est donnée par :

$$F^*(t|\mathcal{H}(u)) = \int_u^t \mathbb{P}(T_{k+1} \in [s + ds]|\mathcal{H}(u))ds = \int_u^t f^*(s|\mathcal{H}(u))ds. \quad (18)$$

La densité jointe d'une réalisation $\{t_1, \dots, t_k\}$ est alors donnée par :

$$f(t_1, t_2, \dots, t_k) = \prod_{i=0}^k f^*(t_i|\mathcal{H}(t_{i-1})). \quad (19)$$

Il est souvent difficile de travailler avec les distributions conditionnelles des temps d'arrivées. Une autre caractérisation des processus ponctuels est utilisée : la fonction d'intensité conditionnelle. Elle est définie comme suit :

Définition 5.1 ([Rasmussen, 2011]). *Soit T un processus ponctuel, $f^*(t|\mathcal{H}(u))$ et $F^*(t|\mathcal{H}(u))$ donnée en (18). On appelle $\lambda^*(.)$ la fonction d'intensité conditionnelle définie par :*

$$\lambda^*(t) = \frac{f^*(t|\mathcal{H}(u))}{1 - F^*(t|\mathcal{H}(u))}. \quad (20)$$

Même si cette définition est valide nous préférons utiliser l'intensité conditionnelle sous la forme suivante :

Définition 5.2 ([Rasmussen, 2011]). *Soit T un processus ponctuel et $\mathcal{H}(t) = \{t_i : t_i \leq t\}$ l'historique du processus. On appelle $\lambda^*(t)$ la fonction définie par :*

$$\lambda^*(t) = \lim_{h \rightarrow 0} \frac{\mathbb{E}[N_{t+h} - N_t|\mathcal{H}(t^-)]}{h}. \quad (21)$$

Propriété 5.1. *Les définitions 5.1 et 5.2 sont équivalentes.*

Démonstration. Cette preuve est celle de [Rasmussen, 2011].

$$\begin{aligned} \lambda^*(t)dt &= \frac{f^*(t|\mathcal{H}(t_n))}{1 - F^*(t|\mathcal{H}(t_n))} \\ &= \frac{\mathbb{P}(t_{n+1} \in [t, t + dt]|\mathcal{H}(t_n))}{\mathbb{P}(t_{n+1} \notin [t_n, t]|\mathcal{H}(t_n))} \\ &= \frac{\mathbb{P}(t_{n+1} \in [t, t + dt], t_{n+1} \notin [t_n, t]|\mathcal{H}(t_n))}{\mathbb{P}(t_{n+1} \notin [t_n, t]|\mathcal{H}(t_n))} \\ &= \mathbb{P}(t_{n+1} \in [t, t + dt]|t_{n+1} \notin [t_n, t], \mathcal{H}(t_n)) \\ &= \mathbb{P}(t_{n+1} \in [t, t + dt]|\mathcal{H}(t^-)). \end{aligned}$$

Si le processus est simple alors :

$$\lambda^*(t)dt = \mathbb{E}[N([t, t + dt])|\mathcal{H}(t^-)].$$

□

La quatrième ligne de cette démonstration donne une interprétation de l'intensité conditionnelle. Elle signifie que $\lambda^*(t)dt$ est égale à la probabilité que le point t_{n+1} apparaisse dans l'intervalle de temps $[t, t + dt]$ en sachant qu'il n'est pas apparu entre l'instant t_n et l'instant t , et en connaissant l'historique du processus.

Il est également possible d'exprimer les fonctions de répartition $F^*(t|\mathcal{H}_{t_k})$ et de densité $f^*(t|\mathcal{H}_{t_k})$ à partir de cette intensité conditionnelle. La proposition suivante relie ces notions.

Proposition 5.1. *[Rasmussen, 2011] Les relations inverses de (20) sont :*

$$F^*(t|\mathcal{H}_{t_k}) = 1 - \exp\left(-\int_{t_k}^t \lambda^*(s)ds\right) \quad (22)$$

$$f^*(t|\mathcal{H}_{t_k}) = \lambda^*(t) \exp\left(-\int_{t_k}^t \lambda^*(s)ds\right) \quad (23)$$

ou t_k est le dernier instant d'arrivée avant l'instant t .

Démonstration. On cherche tout d'abord l'expression de la fonction de répartition :

$$\lambda^*(t) = \frac{f^*(t|\mathcal{H}_{t_k})}{1 - F^*(t|\mathcal{H}_{t_k})} = \frac{\frac{d}{dt}F^*(t|\mathcal{H}_{t_k})}{1 - F^*(t|\mathcal{H}_{t_k})} = -\frac{d}{dt}(\log(1 - F^*(t|\mathcal{H}_{t_k}))).$$

En intégrant des deux côtés cela conduit à :

$$F^*(t|\mathcal{H}_{t_k}) = 1 - \exp\left(-\int_{t_k}^t \lambda^*(s)ds\right).$$

En dérivant cette expression il vient :

$$f^*(t|\mathcal{H}_{t_k}) = \lambda^*(t) \exp\left(-\int_{t_k}^t \lambda^*(s)ds\right).$$

□

En remplaçant $f^*(t|\mathcal{H}_{t_k})$ par la formule (23), la densité jointe des instants d'arrivées (19) est égale à :

$$f(t_1, t_2, \dots, t_k) = \prod_{i=0}^k \lambda^*(t_i) \exp\left(-\int_0^{t_k} \lambda^*(s)ds\right). \quad (24)$$

La propriété suivante permet de caractériser le processus ponctuel à partir de l'intensité conditionnelle.

Proposition 5.2. *[Rasmussen, 2011]*

L'intensité conditionnelle $\lambda^(t)$ définit le processus ponctuel de manière unique si les deux hypothèses suivantes sont réunies pour toute réalisation (t_1, \dots, t_k) et pour tout $t > t_k$:*

1. $\lambda^*(t)$ est non-négative et intégrable sur tout intervalle commençant à t_k .
2. $\int_0^t \lambda^*(s)ds \rightarrow \infty$ quand $t \rightarrow \infty$.

Démonstration. De [Rasmussen, 2011].

Un processus ponctuel est bien défini si les fonctions de répartitions $F^*(t|\mathcal{H}(t_k))$ le sont également, c'est ce que nous allons vérifier.

On s'intéresse à la fonction de répartition du $(k+1)$ -ième point en connaissant l'historique $\mathcal{H}(t|\mathcal{H}(t_k))$. Or d'après l'équation (23) et des hypothèses de l'énoncé il vient :

$$\begin{cases} (i). & 0 \leq F^*(t|\mathcal{H}(t_k)) \leq 1 \\ (ii). & F^*(t|\mathcal{H}(t_k)) \text{ est une fonction non décroissante de } t \\ (iii). & \lim_{t \rightarrow +\infty} F^*(t|\mathcal{H}(t_k)) = 1. \end{cases}$$

□

Cette propriété est centrale dans l'étude des processus de comptage car elle relie de manière unique le processus et la fonction d'intensité conditionnelle.

5.2 Définitions et propriétés du processus de Hawkes

Le processus de Hawkes est un processus ponctuel qui a été introduit par [Hawkes, 1971]. C'est un processus ponctuel défini par une intensité conditionnelle. Le champ d'application est plus vaste que les processus de Poisson car les événements du processus de Hawkes peuvent être liés les uns aux autres. Une des premières utilisation du modèle a été faite par [Ogata, 1988] un mathématicien japonais, pour modéliser les occurrences des tremblements de Terre. Maintenant il est utilisé dans le domaine de la finance [Omi et al., 2017] et [Da Fonseca and Zaatour, 2014] pour suivre l'évolution d'un actif financier ou en épidémiologie [Schoenberg et al., 2019].

Définition 5.3. *Un processus de Hawkes sur \mathbb{R}^+ est un processus de comptage $(N_t)_{t \geq 0}$ défini par une intensité conditionnelle de la forme :*

$$\lambda^*(t) = \lambda(t) + \int_0^t \mu(t-u) dN_u = \lambda(t) + \sum_{t_i < t} \mu(t-t_i) \quad (25)$$

avec $\lambda(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ l'intensité d'arrière plan, $\mu(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ la fonction d'excitation, $(t_i)_{i \geq 1}$ les instants d'arrivées et dN_u la mesure de comptage.

Définition 5.4. *On appelle compensateur du processus de Hawkes l'intégrale de la fonction d'intensité :*

$$\Lambda^*(t) = \int_0^t \lambda^*(s) ds. \quad (26)$$

Remarque 5.1. *Si $\mu(\cdot)$ est identiquement nulle, alors le processus de Hawkes est un processus de Poisson non homogène d'intensité $\lambda(\cdot)$. C'est pourquoi le processus de Hawkes peut être vu comme une extension du processus de Poisson.*

La fonction d'excitation $\mu(\cdot)$ augmente l'intensité $\lambda^*(\cdot)$ dès qu'un nouvel instant d'arrivée t_i apparaît. Ainsi la survenue d'événements augmente la probabilité que d'autres événements arrivent, ils auto-excitent le processus.

Il a par exemple été montré que pour un réseau social, la réaction d'une personne stimule les réactions d'autres utilisateurs.

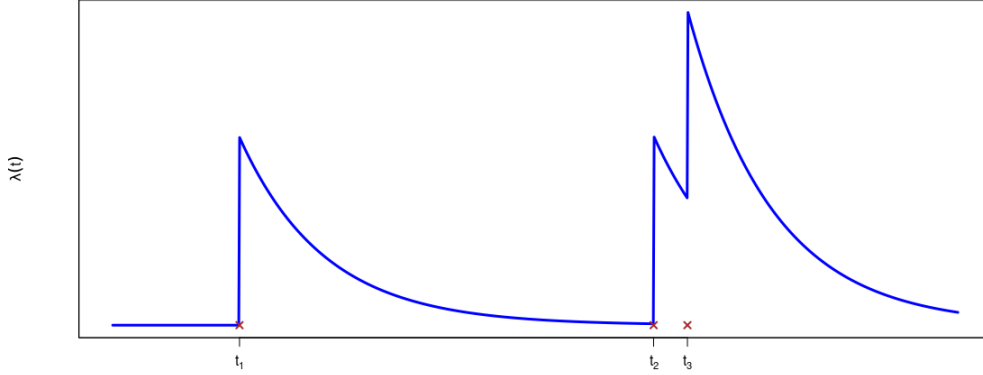


FIGURE 7 – Evolution de l'intensité conditionnelle (en bleu) avec trois instants d'arrivées t_1, t_2 et t_3 (marrons). Sur $(0, t_1]$, $\lambda^*(t)$ est égale à λ . Lorsque le premier point t_1 apparaît, $\lambda^*(t)$ prend la valeur $\lambda + \alpha e^{-\beta(t-t_1)}$ sur l'intervalle $(t_1, t_2]$, etc.

Dans la suite de ce rapport on va considérer une intensité d'arrière plan constante et une fonction d'excitation exponentielle définie par

$$\mu : t \rightarrow \alpha e^{-\beta t} \quad (27)$$

avec $\alpha, \beta > 0$.

Dans ce cas l'intensité conditionnelle s'écrit :

$$\lambda^*(t) = \lambda + \sum_{t_i < t} \alpha e^{-\beta(t-t_i)}. \quad (28)$$

Avec λ, α et $\beta > 0$.

On remarque que la fonction d'excitation $\mu(\cdot)$ augmente l'intensité localement. C'est la raison pour laquelle certains instants d'arrivée sont proches les uns des autres et créent des groupes. Ce phénomène de cluster est caractéristique des processus de Hawkes.

La fonction d'intensité conditionnelle pour trois instants d'arrivées d'un processus de Hawkes est représentée sur la Figure 7.

On désigne par n le facteur de branchement, il est défini par :

$$n = \int_0^\infty \mu(s) ds.$$

On peut interpréter n comme le nombre moyen d'événements qu'un événement va générer par la fonction d'excitation. Pour une fonction d'excitation exponentielle, il est égal à $n = \frac{\alpha}{\beta}$.

On constate que pour connaître $\lambda^*(t)$ il faut connaître les instants d'arrivées (t_i) qui se sont produits avant t . L'intensité $\lambda^*(t)$ devient alors aléatoire et non pas déterministe comme pour les processus de Poisson.

La propriété suivante donne les conditions pour lesquelles $\mathbb{E}(\lambda^*(t))$ est finie.

Propriété 5.2. *On considère un processus de Hawkes avec une intensité d'arrière plan constante λ et fonction d'excitation exponentielle $\mu(\cdot)$ (27). On note $n = \frac{\alpha}{\beta}$ le facteur de branchement. Alors :*

$$\lim_{t \rightarrow \infty} \mathbb{E}[\lambda^*(t)] = \begin{cases} \frac{\lambda}{1-n} & \text{si } \alpha < \beta \\ \infty & \text{sinon.} \end{cases}$$

Démonstration. Par l'expression (21) il vient :

$$\lambda^*(t) = \frac{\mathbb{E}[d(N_t)|\mathcal{H}(t^-)]}{dt}.$$

On prend l'espérance des deux côtés :

$$\mathbb{E}[\lambda^*(t)] = \frac{\mathbb{E}[d(N_t)]}{dt}.$$

Ce qui peut être noté comme :

$$\mathbb{E}[\lambda^*(t)]dt = \mathbb{E}[d(N_t)]. \quad (29)$$

On utilise l'expression de l'intensité conditionnelle d'un processus de Hawkes (25) dont on prend l'espérance :

$$\begin{aligned} \lambda^*(t) &= \lambda + \int_0^t \mu(t-u) dN_u \\ \mathbb{E}[\lambda^*(t)] &= \mathbb{E} \left[\lambda + \int_0^t \mu(t-u) dN_u \right] \\ &= \lambda + \int_0^t \mu(t-u) \mathbb{E}[dN_u]. \end{aligned}$$

En utilisant (29) il vient :

$$\mathbb{E}[\lambda^*(t)] = \lambda + \int_0^t \mu(t-u) \mathbb{E}[\lambda^*(u)] du.$$

Si on note $g(t) = \mathbb{E}[\lambda^*(t)]$ l'équation précédente devient :

$$g(t) = \lambda + \mu * g(t) \quad (30)$$

où $*$ est l'opérateur de convolution.

Trouver $g(t)$ revient à résoudre cette équation de convolution. Pour le faire nous allons utiliser la transformée de Laplace désignée par \mathcal{L} .

L'opérateur \mathcal{L} est linéaire et la transformée de Laplace d'un produit de convolution de deux fonctions est égale au produit des transformées de Laplace de chaque fonction.

Soit $p \in \mathbb{C}$ tel que $\operatorname{Re}(p) > -\beta$ alors $\mathcal{L}(\mu)(p)$ est donnée par :

$$\mathcal{L}(\mu)(p) = \int_0^{+\infty} \mu(t)e^{-tp}dt = \frac{\alpha}{\beta + p}.$$

Soit $p \in \mathbb{C}$ tel que $\operatorname{Re}(p) > -0$, alors la transformée de Laplace de la constante λ est :

$$\mathcal{L}(\lambda)(p) = \int_0^{+\infty} \lambda e^{-tp}dt = \frac{\lambda}{p}.$$

On utilise la transformée de Laplace sur la fonction $g(t)$:

$$\begin{aligned} \mathcal{L}(g)(p) &= \mathcal{L}(\lambda)(p) + \mathcal{L}(\mu * g)(p) \\ &= \mathcal{L}(\lambda)(p) + \mathcal{L}(\mu)(p)\mathcal{L}(g)(p) \\ &= \frac{\mathcal{L}(\lambda)(p)}{1 - \mathcal{L}(\mu)(p)} \\ &= \frac{\lambda(p + \beta)}{p(p + \beta - \alpha)} \\ &= \frac{\lambda}{p} + \frac{\alpha\lambda}{p(p + \beta - \alpha)}. \end{aligned}$$

On souhaite utiliser la transformée de Laplace inverse. Pour se ramener à des expressions de transformée de Laplace usuelles, on réalise une décomposition en éléments simples de $\frac{1}{p(p + \beta - \alpha)}$.

On distingue alors deux cas : $\begin{cases} \text{Cas 1 : } \alpha \neq \beta \\ \text{Cas 2 : } \alpha = \beta \end{cases}$

Si $\alpha \neq \beta$, on réalise une décomposition en éléments simples de $\frac{1}{p(p + \beta - \alpha)}$:

$$\frac{1}{p(p + \beta - \alpha)} = \frac{1}{(\beta - \alpha)p} - \frac{1}{(\beta - \alpha)(p + \beta - \alpha)}.$$

On en déduit l'expression de $\mathcal{L}(g)(p)$:

$$\begin{aligned} \mathcal{L}(g)(p) &= \frac{\lambda}{p} + \frac{\alpha\lambda}{(\beta - \alpha)p} - \frac{\alpha\lambda}{(\beta - \alpha)(p + \beta - \alpha)} \\ &= \frac{\beta\lambda}{\beta - \alpha} \frac{1}{p} - \frac{\alpha\lambda}{(\beta - \alpha)} \frac{1}{(p + \beta - \alpha)}. \end{aligned}$$

Si $\alpha = \beta$, alors il y a une racine double :

$$\frac{1}{p(p + \beta - \alpha)} = \frac{1}{p^2}.$$

On en déduit que :

$$\mathcal{L}(g)(p) = \frac{\lambda}{p} + \frac{\alpha\lambda}{p^2}.$$

Ainsi

$$\mathcal{L}(g)(p) = \begin{cases} \frac{\beta\lambda}{\beta-\alpha} \frac{1}{p} - \frac{\alpha\lambda}{\beta-\alpha} \frac{1}{p+(\beta-\alpha)} & \text{si } \alpha \neq \beta \\ \frac{\lambda}{p} + \frac{\alpha\lambda}{p^2} & \text{si } \alpha = \beta. \end{cases}$$

On passe ensuite par la transformée de Laplace inverse pour retrouver la fonction $g(t)$. L'opérateur \mathcal{L}^{-1} est linéaire, et on a les résultats suivants : $\mathcal{L}^{-1}(\frac{1}{p}) = 1$, $\mathcal{L}^{-1}(\frac{1}{p^2}) = t$ et $\mathcal{L}^{-1}(\frac{1}{p+(\beta-\alpha)}) = e^{-(\beta-\alpha)t}$ lorsque $\beta \neq \alpha$. On en déduit que :

$$g(t) = \begin{cases} \frac{\beta\lambda}{\beta-\alpha} - \frac{\alpha\lambda}{\beta-\alpha} e^{-(\beta-\alpha)t} & \text{si } \alpha \neq \beta \\ \lambda + \alpha\lambda t & \text{si } \alpha = \beta. \end{cases}$$

On pose $n = \int_0^{+\infty} \alpha e^{-\beta t} dt = \frac{\alpha}{\beta}$ et on fait tendre t vers l'infini pour obtenir :

$$\lim_{t \rightarrow \infty} \mathbb{E}[\lambda^*(t)] = \lim_{t \rightarrow \infty} g(t) = \begin{cases} \frac{\lambda}{1-n} & \text{si } \alpha < \beta \\ \infty & \text{sinon.} \end{cases}$$

□

Lorsque la condition $\alpha < \beta$ est satisfaite, l'espérance tend vers une constante positive finie. Dans ce cas, le facteur de branchement $n = \frac{\alpha}{\beta}$ peut être interprété comme la contribution des événements du système eux mêmes. Plus il est proche de 1, plus cette contribution est grande. Elle est même hyperbolique car l'espérance est proportionnelle à $\frac{1}{1-n}$.

La Figure 8 montre l'évolution de l'intensité conditionnelle d'un processus de Hawkes.

5.3 Simulation

Dans cette section, trois méthodes de simulation du processus de Hawkes sont présentées : l'algorithme d'amincissement modifié d'Ogata, la méthode de superposition et la méthode de l'inverse du compensateur.

5.3.1 Algorithme d'amincissement modifié d'Ogata

Pour simuler un processus de Hawkes, on peut utiliser une procédure d'amincissement. On simule un grand nombre de points et on procède par acceptation rejet pour déterminer quels points vont être conservés. Cette procédure introduite par [Ogata, 1981] se base sur l'idée que l'on peut toujours trouver une borne supérieure locale de l'intensité conditionnelle. En effet on ne connaît pas à priori de borne supérieure pour l'intensité, mais la fonction est déterministe et décroissante entre deux instants d'arrivées. La preuve mathématique de l'amincissement modifié est donné dans [Allal, 2020].

Le pseudo code est présenté par l'Algorithme 6 [Laub et al., 2015]. Le code **R** est donné en annexe 8.8.

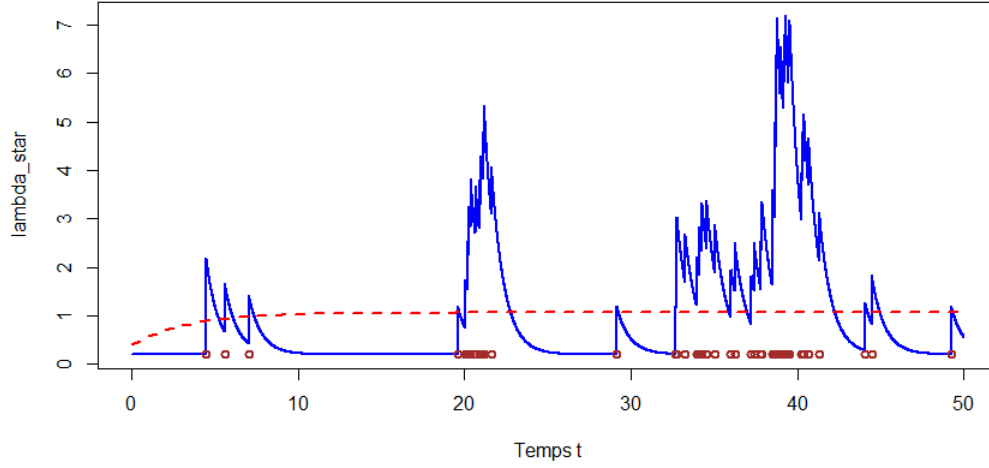


FIGURE 8 – Réalisation d'un processus de Hawkes avec les paramètres $\lambda = 0.2, \alpha = 1, \beta = 1.3, T = 50$. L'intensité conditionnelle est représentée en bleue l'espérance de l'intensité conditionnelle en rouge pointillé et les instants d'arrivée en marron. On constate que les instants d'arrivées sont groupés.

Algorithm 6 Simulation du processus de Hawkes par amincissement d'Ogata.

Require: $\lambda^*(.)$ et T

- 1: $Liste \leftarrow [], t \leftarrow 0$
- 2: $\epsilon \leftarrow 10^{-10}$
- 3: **while** $t < T$ **do**
- 4: $M \leftarrow \lambda^*(t + \epsilon)$ On actualise la borne supérieure à la valeur de l'intensité
- 5: $W \leftarrow \mathcal{E}(M)$ On simule une loi exponentielle de paramètre la borne supérieure
- 6: $t \leftarrow t + W$
- 7: $U \leftarrow \text{Unif}(0, M)$
- 8: **if** $U \leq \lambda^*(t)$ **and** $t < T$ **then**
- 9: $Liste \leftarrow [Liste, t]$
- 10: **end if**
- 11: **end while**
- 12: **return** $Liste$

On a besoin d'avoir une borne supérieure pour utiliser une procédure d'amincissement. Ogata propose d'actualiser la borne supérieure à chaque étape. Cela consiste à générer une loi exponentielle de paramètre la borne actuelle, puis de réaliser une procédure d'acceptation rejet à partir de la position de l'intensité conditionnelle et d'une simulation de loi uniforme. La borne supérieure est actualisée en fonction de l'acceptation ou du rejet. Si le point est accepté alors la borne supérieure

prend la valeur de la fonction d'intensité conditionnelle en sachant qu'elle fait un bond d'amplitude α . Si le point est refusé, la borne supérieure est actualisée à la valeur de l'intensité conditionnelle à ce point. Ce qui est possible car la fonction d'intensité conditionnelle est décroissante entre deux sauts.

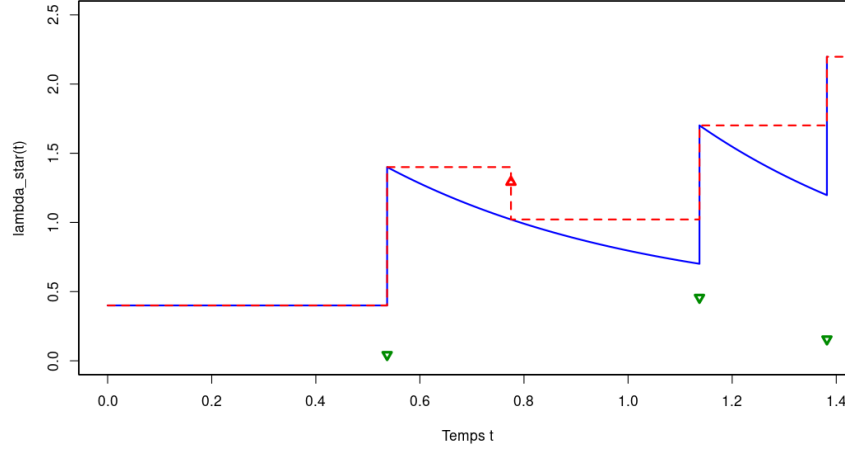


FIGURE 9 – Procédure d'amincissement d'Ogata. La courbe bleue est l'intensité conditionnelle, la droite rouge en pointillés est la borne supérieure locale, les points verts sous la courbe sont les points acceptés et le point rouge est le point refusé.

5.3.2 Simulation par l'inverse du compensateur

On peut aussi simuler un processus de Hawkes en utilisant les points d'un processus de Poisson homogène. On les transforme avec ce que l'on appelle l'inverse du compensateur comme le montre la propriété suivante :

Propriété 5.3. *On suppose que $\lambda^*(.)$ est une intensité conditionnelle d'un processus ponctuel tel que pour tout $t \geq 0$, $\lambda^*(t) > 0$ et $\lim_{t \rightarrow +\infty} \Lambda^*(t) = +\infty$.*

On considère $(s_i)_{i \in \mathbb{N}}$ une réalisation d'un processus de Poisson d'intensité unitaire sur \mathbb{R}^+ et $t_i = \Lambda^{-1}(s_i)$, alors $(t_i)_{i \in \mathbb{N}}$ forment un processus d'intensité $\lambda^*(.)$.*

Remarque 5.2. *Cette propriété est valable pour n'importe quel processus ponctuel simple. Il s'agit de la même propriété que l'inverse du compensateur pour les processus de Poisson non homogène 4.3 étendue à des intensités conditionnelles. La démonstration est identique.*

Ainsi, il revient à rechercher le prochain point t_{k+1} tel que $t_{k+1} = \Lambda^{*-1}(s_{k+1})$ avec $(s_i)_{i \geq 1}$ une réalisation d'un processus de Poisson homogène d'intensité 1. Or on ne connaît pas d'expression de

Λ^{*-1} , on doit alors chercher le point t_{k+1} autrement. On sait que t_{k+1} satisfait :

$$\int_{t_k}^{t_{k+1}} \lambda^*(u) du = -\ln(U) \text{ avec } U \sim \text{Unif}(0, 1).$$

En considérant une fonction d'excitation exponentielle cela mène à :

$$\ln(U) + \lambda(t_{k+1} - t_k) - \frac{\alpha}{\beta} \left(\sum_{i=1}^k e^{-\beta(t_{k+1}-t_i)} - e^{-\beta(t_k-t_i)} \right) = 0.$$

Les sommes peuvent se calculer de manière récursive en posant $S(k) = \sum_{i=1}^k e^{-\beta(t_k-t_i)}$ qui vérifie

$$S(k) = 1 + e^{-\beta(t_k-t_{k-1})} S(k-1) \text{ et } S(1) = 1.$$

On a ainsi :

$$\ln(U) + \lambda(t_{k+1} - t_k) - \frac{\alpha}{\beta} \left(S(k)(e^{-\beta(t_{k+1}-t_k)} - 1) \right) = 0.$$

Cette recherche du point t_{k+1} peut se faire de manière itérative en annulant une fonction à chaque point. Le pseudo code associé est présenté dans l'Algorithme 7. Le code **R** est fourni en annexe 8.7.

Algorithm 7 [Ozaki, 1979] Simulation du processus de Hawkes par inversion

Require: Le compensateur $\Lambda(\cdot)$, une borne de temps T

```

1: Liste  $\leftarrow \emptyset$ 
2:  $t_k \leftarrow 0$  le dernier point de Liste, 0 initialement
3: while  $t_k \leq T$  do
4:    $U \leftarrow \text{Unif}(0, 1)$  simulation d'une loi uniforme sur l'intervalle de temps  $[0, 1]$ .
5:   Trouver  $t$  tel que  $\ln(U) + \lambda(t - t_k) - \frac{\alpha}{\beta} (S(k)(e^{-\beta(t-t_k)} - 1)) = 0$ 
6:    $t_k \leftarrow t$ 
7:   if Si  $t_k \leq T$  then
8:     Ajouter  $t_k$  à Liste
9:   end if
10: end while
11: return Liste
```

5.4 Inférence statistique

Dans cette section, l'estimateur du maximum de vraisemblance pour le processus de Hawkes et la validation du modèle sont présentés.

5.4.1 Vraisemblance du modèle

On choisit d'estimer les paramètres à l'aide de l'estimateur du maximum de vraisemblance défini dans la propriété suivante.

Propriété 5.4. Soit une réalisation (t_1, \dots, t_k) d'un processus de Hawkes sur un intervalle de temps $(0, T]$, alors la fonction de vraisemblance du processus du paramètre θ est donnée par :

$$L(\theta|t_1, \dots, t_k, T) = \prod_{i=1}^k \lambda_{\theta}^*(t_i) \exp\left(-\int_0^T \lambda_{\theta}^*(u) du\right). \quad (31)$$

Démonstration. Voir [Daley and Vere-Jones, 2003] Proposition 7.2.III. \square

Propriété 5.5. *On suppose que l'intensité conditionnelle a une intensité de background λ constante et une fonction d'excitation exponentielle, c'est-à-dire $\theta = (\lambda, \alpha, \beta)$. Alors en se limitant à $T = t_k$, on peut avoir une expression simplifiée de la log-vraisemblance donnée par :*

$$l(\theta|t_1, \dots, t_k) = \sum_{i=1}^k \ln(\lambda + \alpha A(i)) - \lambda t_k + \frac{\alpha}{\beta} \sum_{i=1}^k \left[e^{-\beta(t_k - t_i)} - 1 \right] \quad (32)$$

avec $A(i) = \sum_{j=1}^{i-1} e^{-\beta(t_i - t_j)}$ pour $i > 1$ et $A(1) = 0$.

En utilisant cette expression, il est possible de calculer le gradient ainsi que la matrice Hessienne de la log-vraisemblance. Pour le gradient on a les expressions suivantes :

$$\begin{aligned} \frac{\partial l}{\partial \lambda} &= -t_k + \sum_{i=1}^k \frac{1}{\lambda + \alpha A(i)} \cdot \frac{\partial l}{\partial \alpha} = \sum_{i=1}^k \frac{A(i)}{\lambda + \alpha A(i)} + \frac{1}{\beta} \sum_{i=1}^k \left[e^{-\beta(t_k - t_i)} - 1 \right] . \\ \frac{\partial l}{\partial \beta} &= \sum_{i=1}^k \frac{-\alpha B(i)}{\lambda + \alpha A(i)} - \alpha \sum_{i=1}^k \left[\frac{1}{\beta} (t_k - t_i) e^{-\beta(t_k - t_i)} + \frac{1}{\beta^2} (e^{-\beta(t_k - t_i)} - 1) \right] \end{aligned}$$

avec $B(i) = \sum_{j=1}^{i-1} (t_i - t_j) e^{-\beta(t_i - t_j)}$ pour $i > 1$ et $B(1) = 0$.

Les éléments de la matrice Hessienne de la log-vraisemblance sont :

$$\begin{aligned} \frac{\partial^2 l}{\partial \lambda^2} &= - \sum_{i=1}^k \frac{1}{(\lambda + \alpha A(i))^2} . \\ \frac{\partial^2 l}{\partial \lambda \partial \alpha} &= - \sum_{i=1}^k \frac{A(i)}{(\lambda + \alpha A(i))^2} . \\ \frac{\partial^2 l}{\partial \lambda \partial \beta} &= \sum_{i=1}^k \frac{\alpha B(i)}{(\lambda + \alpha A(i))^2} . \\ \frac{\partial^2 l}{\partial \alpha^2} &= - \sum_{i=1}^k \left(\frac{A(i)}{\lambda + \alpha A(i)} \right)^2 . \\ \frac{\partial^2 l}{\partial \alpha \partial \beta} &= \sum_{i=1}^k \frac{-B(i)}{(\lambda + \alpha A(i))^2} + \frac{\alpha A(i) B(i)}{(\lambda + \alpha A(i))^2} - \frac{1}{\beta^2} (e^{-\beta(t_k - t_i)} - 1) - \frac{1}{\beta} (t_k - t_i) e^{-\beta(t_k - t_i)} . \\ \frac{\partial^2 l}{\partial \beta^2} &= \sum_{i=1}^k \frac{\alpha C(i)}{\lambda + \alpha A(i)} - \left(\frac{\alpha B(i)}{\lambda + \alpha A(i)} \right)^2 \\ &\quad + \alpha \left[\frac{2}{\beta^3} (e^{-\beta(t_k - t_i)} - 1) + \frac{2}{\beta^2} (t_k - t_i) e^{-\beta(t_k - t_i)} + \frac{1}{\beta} (t_k - t_i)^2 e^{-\beta(t_k - t_i)} \right] . \end{aligned}$$

On note $C(i) = \sum_{j=1}^{i-1} (t_i - t_j)^2 e^{-\beta(t_i - t_j)}$ et $C(1) = 0$.

Les $A(i)$, $B(i)$ et $C(i)$ se calculent récursivement avec

1. $A(i) = e^{-\beta(t_i - t_{i-1})}(1 + A(i-1))$ pour $i > 1$, et $A(1) = 0$.
2. $B(i) = e^{-\beta(t_i - t_{i-1})}[(t_i - t_{i-1}) + (t_i - t_{i-1})A(i-1) + B(i-1)]$ pour $i > 1$ et $B(1) = 0$.
3. $C(i) = e^{-\beta(t_i - t_{i-1})}[(t_i - t_{i-1})^2 + (t_i - t_{i-1})^2 A(i-1) + 2(t_i - t_{i-1})B(i-1) + C(i-1)]$ pour $i > 1$ et $C(1) = 0$.

Démonstration. On montre comment obtenir la \log -vraisemblance l dans le cas d'une fonction d'excitation exponentielle, ainsi que les expressions récursives des $A(i)$, $B(i)$ et $C(i)$.

En se limitant à $T = t_k$ l'expression de la vraisemblance est :

$$L(\theta|t_1, \dots, t_k) = \prod_{i=1}^k \lambda^*(t_i) \cdot \exp\left(-\int_0^{t_k} \lambda^*(s) ds\right).$$

On passe à la \log -vraisemblance pour obtenir :

$$l(\theta|t_1, \dots, t_k) = \sum_{i=1}^k \ln(\lambda^*(t_i)) - \int_0^{t_k} \lambda^*(s) ds.$$

L'intensité conditionnelle $\lambda^*(.)$ a une fonction d'excitation exponentielle d'où :

$$\lambda^*(t_i) = \lambda + \sum_{j=1}^{i-1} \alpha \exp(-\beta(t - t_j)).$$

On pose $A(i) = \sum_{j=1}^{i-1} \exp(-\beta(t_i - t_j))$ pour $i > 1$, et $A(1) = 0$ sinon, on obtient ainsi :

$$\lambda^*(t_i) = \lambda + \alpha A(i).$$

Le second terme de cette \log -vraisemblance est le compensateur. Son expression est donnée par :

$$\begin{aligned} \Lambda^*(t) &= \int_0^t \lambda + \sum_{t_i < s} \alpha \exp(-\beta(s - t_i)) ds \\ &= \lambda t + \sum_{t_i < t} \int_{t_i}^t \alpha \exp(-\beta(s - t_i)) ds \\ &= \lambda t + \frac{\alpha}{\beta} \sum_{t_i < t} (1 - \exp(-\beta(t - t_i))). \end{aligned}$$

On en déduit l'expression de la \log -vraisemblance :

$$l(\theta|t_1, \dots, t_k) = \sum_{i=1}^k \ln(\lambda + \alpha A(i)) - \lambda t_k + \frac{\alpha}{\beta} \sum_{i=1}^k (\exp(-\beta(t_k - t_i)) - 1).$$

□

Les formes récursives des $A(i)$, $B(i)$ et $C(i)$ sont les suivantes :

Démonstration.

$$\begin{aligned}
A(i) &= \sum_{j=1}^{i-1} e^{-\beta(t_i - t_j)} \\
&= e^{-\beta(t_i - t_{i-1})} \sum_{j=1}^{i-1} e^{-\beta(t_{i-1} - t_j)} \\
&= e^{-\beta(t_i - t_{i-1})} \left[1 + \sum_{j=1}^{i-2} e^{-\beta(t_{i-1} - t_j)} \right].
\end{aligned}$$

Donc on obtient finalement :

$$A(i) = e^{-\beta(t_i - t_{i-1})} [1 + A(i-1)].$$

On s'intéresse maintenant aux $B(i)$. L'idée consiste à exprimer $B(i)$ en fonction de $B(i-1)$ et de $A(i-1)$:

$$\begin{aligned}
B(i) &= \sum_{j=1}^{i-1} (t_i - t_j) e^{-\beta(t_i - t_j)} \\
&= e^{-\beta(t_i - t_{i-1})} \sum_{j=1}^{i-1} (t_i - t_j) e^{-\beta(t_{i-1} - t_j)} \\
&= e^{-\beta(t_i - t_{i-1})} \left[(t_i - t_{i-1}) + \sum_{j=1}^{i-2} (t_i - t_j) e^{-\beta(t_{i-1} - t_j)} \right].
\end{aligned}$$

On décompose le terme de droite. On retire le terme en t_i et on injecte un terme en t_{i-1} pour retrouver $B(i-1)$:

$$\begin{aligned}
\sum_{j=1}^{i-2} (t_i - t_j) e^{-\beta(t_{i-1} - t_j)} &= \sum_{j=1}^{i-2} t_i e^{-\beta(t_{i-1} - t_j)} - \sum_{j=1}^{i-2} t_j e^{-\beta(t_{i-1} - t_j)} + \sum_{j=1}^{i-2} t_{i-1} e^{-\beta(t_{i-1} - t_j)} \\
&\quad - \sum_{j=1}^{i-2} t_{i-1} e^{-\beta(t_{i-1} - t_j)} \\
&= (t_i - t_{i-1}) A(i-1) + B(i-1).
\end{aligned}$$

On en déduit que :

$$B(i) = e^{-\beta(t_i - t_{i-1})} [(t_i - t_{i-1}) + (t_i - t_{i-1}) A(i-1) + B(i-1)].$$

On procède de la même façon pour trouver l'expression récursive des $C(i)$. Nous allons exprimer $C(i)$ en fonction de $A(i-1)$, $B(i-1)$ et $C(i-1)$:

$$\begin{aligned} C(i) &= \sum_{j=1}^{i-1} (t_i - t_j)^2 e^{-\beta(t_i - t_j)} \\ &= e^{-\beta(t_i - t_{i-1})} \sum_{j=1}^{i-1} (t_i - t_j)^2 e^{-\beta(t_{i-1} - t_j)} \\ &= e^{-\beta(t_i - t_{i-1})} \left[(t_i - t_{i-1})^2 + \sum_{j=1}^{i-2} (t_i - t_j)^2 e^{-\beta(t_{i-1} - t_j)} \right]. \end{aligned}$$

On décompose le terme $\sum_{j=1}^{i-2} (t_i - t_j)^2 e^{-\beta(t_{i-1} - t_j)}$:

$$\sum_{j=1}^{i-2} (t_i - t_j)^2 e^{-\beta(t_{i-1} - t_j)} = t_i^2 A(i-1) - 2 \sum_{j=1}^{i-2} t_i t_j e^{-\beta(t_{i-1} - t_j)} + \sum_{j=1}^{i-2} t_j^2 e^{-\beta(t_{i-1} - t_j)}.$$

En utilisant la même décomposition que pour les $B(i)$, $\sum_{j=1}^{i-2} t_i t_j e^{-\beta(t_{i-1} - t_j)}$ peut s'écrire :

$$\begin{aligned} \sum_{j=1}^{i-2} t_i t_j e^{-\beta(t_{i-1} - t_j)} &= t_i \sum_{j=1}^{i-2} t_j e^{-\beta(t_{i-1} - t_j)} - t_i \sum_{j=1}^{i-2} t_{i-1} e^{-\beta(t_{i-1} - t_j)} + t_i \sum_{j=1}^{i-2} t_{i-1} e^{-\beta(t_{i-1} - t_j)} \\ &= -t_i B(i-1) + t_i t_{i-1} A(i-1). \end{aligned}$$

Ainsi :

$$\sum_{j=1}^{i-2} (t_i - t_j)^2 e^{-\beta(t_{i-1} - t_j)} = (t_i^2 - 2t_i t_{i-1}) A(i-1) + 2t_i B(i-1) + \sum_{j=1}^{i-2} t_j^2 e^{-\beta(t_{i-1} - t_j)}.$$

On s'intéresse à ce dernier terme en souhaitant faire apparaître le terme $C(i-1)$. On injecte un terme en t_{i-1}^2 et un terme en $t_{i-1} t_j$ pour faire apparaître un terme en $(t_j - t_{i-1})^2$:

$$\begin{aligned} \sum_{j=1}^{i-2} t_j^2 e^{-\beta(t_{i-1} - t_j)} &= \sum_{j=1}^{i-2} t_j^2 e^{-\beta(t_{i-1} - t_j)} + \sum_{j=1}^{i-2} t_{i-1}^2 e^{-\beta(t_{i-1} - t_j)} - 2 \sum_{j=1}^{i-2} t_{i-1} t_j e^{-\beta(t_{i-1} - t_j)} \\ &\quad - \sum_{j=1}^{i-2} t_{i-1}^2 e^{-\beta(t_{i-1} - t_j)} + 2 \sum_{j=1}^{i-2} t_{i-1} t_j e^{-\beta(t_{i-1} - t_j)} \\ &= C(i-1) - t_{i-1}^2 A(i-1) + 2 \sum_{j=1}^{i-2} t_{i-1} t_j e^{-\beta(t_{i-1} - t_j)}. \end{aligned}$$

On exprime aussi $\sum_{j=1}^{i-2} t_{i-1} t_j e^{-\beta(t_{i-1}-t_j)}$ en fonction des $A(i-1)$ et des $B(i-1)$:

$$\begin{aligned} \sum_{j=1}^{i-2} t_{i-1} t_j e^{-\beta(t_{i-1}-t_j)} &= t_{i-1} \sum_{j=1}^{i-2} t_j e^{-\beta(t_{i-1}-t_j)} - t_{i-1} \sum_{j=1}^{i-2} t_{i-1} e^{-\beta(t_{i-1}-t_j)} \\ &\quad + t_{i-1} \sum_{j=1}^{i-2} t_{i-1} e^{-\beta(t_{i-1}-t_j)} \\ &= (t_{i-1})^2 A(i-1) - t_{i-1} B(i-1). \end{aligned}$$

On regroupe tout ces termes dans l'équation de départ pour obtenir :

$$C(i) = e^{-\beta(t_i-t_{i-1})} [(t_i - t_{i-1})^2 + A(i-1)(t_i - t_{i-1})^2 + B(i-1)(2(t_i - t_{i-1})) + C(i-1)].$$

□

Les forme récursives des $A(i)$, $B(i)$ et $C(i)$ sont utiles pour la mise en oeuvre numérique. En effet cela permet de garder une complexité en $O(n)$ plutôt qu'en $O(n^2)$ avec des doubles sommes.

Pour rechercher le maximum de vraisemblance, il faudra calculer les $A(i)$ et les $B(i)$ de nombreuses fois. Exécuter de nombreux calculs avec une complexité quadratique peut être très lourd voir infaisable pour une machine. Utiliser les formes récursives dans les algorithmes nous ramène a une complexité linéaire et rend ces opérations accessibles. On s'aperçoit de plus que les $B(i)$ sont fonctions des $A(i)$, et que les $C(i)$ sont fonctions des $A(i)$ et des $B(i)$. Pour la mise en oeuvre numérique, il suffit de calculer successivement les $A(i)$ puis les $B(i)$ et enfin les $C(i)$. Les codes R sont données en annexe 8.9 et 8.10.

Des résultats sur l'estimateur de vraisemblance ont été démontré par [Ogata, 1978] :

THÉORÈME 3. (OGATA 1978) La matrice Hessienne $\frac{1}{T} \frac{\partial^2 l(\theta)}{\partial \theta_i \partial \theta_j} 1 \leq i, j \leq 3$ est asymptotiquement définie négative dans un certain voisinage U de θ_0 .

THÉORÈME 5. (OGATA 1978) Supposons que le maximum de vraisemblance $\hat{\theta}$ satisfait $\frac{\partial l(\hat{\theta})}{\partial \theta} = 0$, alors :

$$\sqrt{T}(\hat{\theta} - \theta_0) \xrightarrow{T \rightarrow \infty} \mathcal{N}(0, I(\theta_0)^{-1}). \quad (33)$$

$$2\{l(\hat{\theta}) - l(\theta_0)\} \xrightarrow{T \rightarrow \infty} \chi_d^2 \quad (34)$$

en loi.

Afin de vérifier numériquement si les estimations sont précises, plusieurs trajectoires d'un processus de Hawkes sont simulées avec les mêmes paramètres et on regarde les paramètres inférés. On constate que l'estimateur du maximum de vraisemblance semble sans biais.

	$\hat{\lambda}$	$\hat{\alpha}$	$\hat{\beta}$
$T = 100$	0.51	1.10	1.78
$T = 300$	0.53	1.02	1.57
$T = 500$	0.52	1.00	1.54

TABLE 1 – Etude de l’influence de T sur l’estimation par maximum de vraisemblance des paramètres λ , α et β du processus de Hawkes.

5.4.2 Validation de modèle

Après avoir inféré des paramètres $\hat{\lambda}, \hat{\alpha}$ et $\hat{\beta}$, on souhaite savoir si les points de départ suivent un processus de Hawkes. Pour valider l’hypothèse selon laquelle la liste de points suit un processus de Hawkes, on peut utiliser la propriété suivante.

Propriété 5.6. *Soit $(t_i)_{i \in \mathbb{N}}$ un processus d’intensité $\lambda^*(.)$, avec $\lambda^*(.)$ strictement positive et $\lim_{t \rightarrow +\infty} \Lambda^*(t) = +\infty$. Alors les points $(s_i)_{i \in \mathbb{N}}$ définis par $s_i = \Lambda^*(t_i)$ forment un processus de Poisson homogène d’intensité unitaire.*

Remarque 5.3. *Cette propriété est démontrée de la même façon que la propriété 4.3.*

Or pour vérifier qu’un processus est un processus de Poisson, il est possible de s’intéresser aux temps d’inter-arrivés. Ces temps d’inter-arrivées sont appelés résidus ils sont définis comme pour les processus de Poisson non homogène :

Définition 5.5. *Soient (t_1, \dots, t_k) une réalisation d’un processus ponctuel d’intensité conditionnelle $\lambda^*(.)$ strictement positive et $\lim_{t \rightarrow +\infty} \Lambda^*(t) = +\infty$. On adjoint à cette liste $t_0 = 0$. Alors les résidus R_i du processus définis par :*

$$R_i = \Lambda^*(t_i) - \Lambda^*(t_{i-1}) \quad (35)$$

pour $1 \leq i \leq k$ forment un échantillon de la loi exponentielle de paramètre 1.

Afin de valider le modèle, il suffit donc de vérifier que les résidus du processus de Hawkes suivent la loi exponentielle de paramètre 1, on parle d’analyse résiduelle. Le code **R** est disponible en annexe 8.11.

Les processus ponctuels ont été étudié dans l’objectif de modéliser la libération de spores du champignon *Venturia Inaequalis*. Dans cette section on présente dans un premier temps les données de libération de spores. Puis on les ajuste aux processus ponctuels étudiés précédemment.

6 Modélisation des données épidémiques

6.1 Piégeage de spores et données météorologique

Dans l’objectif de connaître la dynamique du champignon *Venturia Inaequalis*, la capture des spores et des mesures climatiques ont été menées dans le verger expérimental de l’Institut de



FIGURE 10 – Piège à spores Burkard.

recherche en Pomologie et Floriculture de Skierniewice en Pologne de 2011 à 2012 [Jankowski and Masny, 2020]. La température de l'air (AIRTEMP), les précipitations (TEMP1), l'humidité relative (RELHUMID) et l'humidité des feuilles (LEAFWET) ont été enregistrées par une station météorologique placée sur le site expérimental. Le Tableau 3 donne la liste complète des données climatiques.

Un piège à spores Burkard (Figure 10) a été utilisé pour détecter la libération des ascospores. Environ 2m^2 du sol du verger sous l'échantillonneur Burkard ont été recouverts de feuilles de la variété de pomme McIntosh fortement infectée par le champignon *V. Inaequalis* la saison précédente. Les arbres n'ont pas été traités par des fongicides pendant l'expérience et avant. Les saisons de décharge d'ascospores dans les années 2011 et 2012 ont duré 59 et 63 jours, les dates de la première et de la dernière libération de spores variant entre le 4 avril et le 1 juin en 2011 et entre le 13 avril et le 14 juin en 2012. Les données de comptage de spores sont représentées dans le Tableau 2 et sur la Figure 11.

Les processus ponctuels sont étudiés dans l'objectif de modéliser la libération de spores. Cela permet de connaître la dynamique du champignon *Venturia Inaequalis* et ainsi de traiter les pommiers le mieux possible. On considère le processus ponctuel défini par la suite des instants de libération de spores. On dénombre $k = 215$ instants de libérations de spores sur une période de $T = 1416$ heures pour l'année 2011, et $k = 174$ instants de libérations de spores sur une période de $T = 1512$ heures en 2012. Ces données sont mesurées à raison d'une mesure par heure.

Les données météorologiques sont mesurées à raison de 5 mesures par heures. Afin de traiter les données météorologiques dans la même échelle que les données de libérations de spores, on attribue une mesure par heure en prenant le maximum des cinq mesures sur chaque heure.

Année	Mois	Jour	Heure	Amplitude
2011	4	4	13	1
2011	4	4	14	1
2011	4	4	15	0
2011	4	4	16	0
2011	4	4	17	65

TABLE 2 – Fragment des données de libération de spores dans le temps. Les quatre premières colonnes représentent l'année, le mois, le jour et l'heure d'une mesure. La colonne "Amplitude" désigne une intensité proportionnelle au nombre de spores enregistré par le piège à spores. Cette intensité est un nombre entier qui varie entre 0 et 15000.

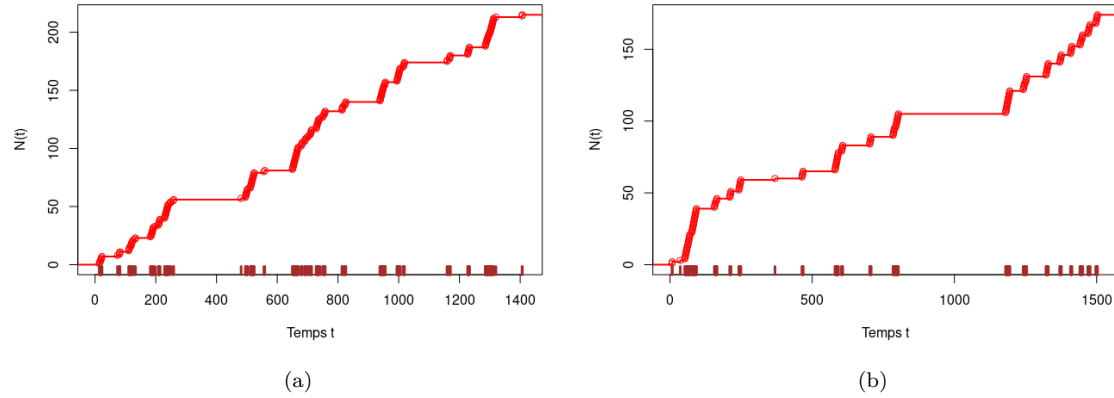


FIGURE 11 – Visualisation des instants de libération de spores pour l'année 2011 (a) et 2012 (b) sous la forme d'un processus de comptage. L'abscisse est mesuré en heure et l'ordonnée $N(t)$ correspond au nombre d'instants de spores enregistrés jusqu'au temps t . Chaque barre sur l'axe des abscisse correspond à un instant de libération de spores.

Variable	Nom	Type	Étendue ou modalité
Température de l'air	AIRTEMP	Quantitatif	$[-1.28, 30.24]$
Humidité de l'air	RELHUMID	Quantitatif	$[24, 99]$
Humidité des feuilles	LEAFWET1	Qualitatif	1 (feuilles humides) 0 sinon
Hauteur des précipitations	PRECIPIT	Quantitatif	$[0, 12.8]$
Intensité lumineuses	SOLAR	Quantitatif	$[0, 879]$
Luminosité	BRIGHTNESS	Qualitatif	1- jour 0- nuit

TABLE 3 – Description des données climatiques utilisées pour la modélisation de la libération de spores dans le temps. Pour chaque variable, le nom, le type et l'étendue ou modalité sont précisés.

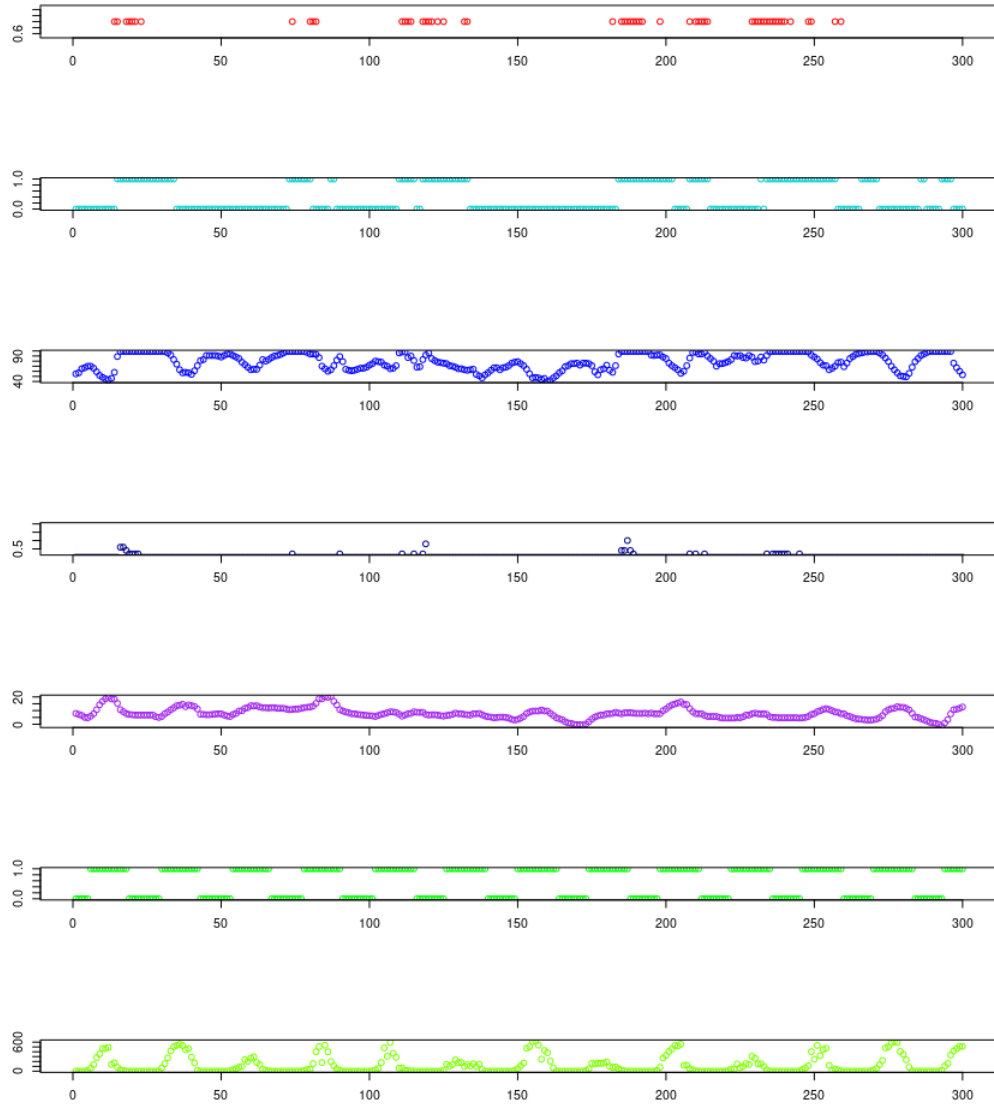


FIGURE 12 – Représentation des instants de libération de spores par un processus ponctuel (premier graphique) et graphiques des données climatiques sur les 300 premières heures de mesures en 2011. Les données climatiques sont : l'humidité des feuilles (2ième graphique), l'humidité relative de l'air (3ième graphique), la hauteur des précipitation (4ième graphique), la température (5ième graphique), la présence de luminosité (6ième graphique) et ensoleillement (7ième graphique). On constate une corrélation entre l'humidité des feuilles et les instants de libérations de spores.

6.2 Ajustement du processus de Poisson non homogène avec une intensité log-linéaire

On ajuste le processus de Poisson non homogène avec une intensité log-linéaire (4.1) aux données. On rappelle que dans ce cas l'intensité s'écrit de la forme :

$$\lambda(t, \theta) = \exp(\theta^\top Z(t)).$$

Avec $\theta = (\theta_1, \dots, \theta_p)$ le vecteur des paramètres et $Z(t) = (Z_1(t), \dots, Z_p(t))$ le vecteur des covariables météorologiques au temps t avec $Z_1(t)$ constante.

Les instants d'arrivées de libérations de spores sont désignés par $(t_i)_{1 \leq i \leq k}$ avec k le nombre d'instantanés d'arrivées. Les deux années sont traitées ensemble en incluant une covariable effet année. On a ainsi $k = 389$ instants de libérations de spores sur une période de $T = 2928$ heures.

La valeur de l'intensité à l'instant d'arrivée t_i est donnée par :

$$\lambda(t_i, \theta) = \exp(\theta^\top Z(t_i)).$$

D'après (15), la log-vraisemblance d'un processus de Poisson non homogène avec une intensité log-linéaire est :

$$l(\theta|t_1, \dots, t_k, T) = \sum_{i=1}^k \ln(\lambda(t_i; \theta)) - \int_0^T \lambda(t; \theta) dt = \sum_{i=1}^k \theta^\top Z(t_i) - \int_0^T \exp(\theta^\top Z(t)) dt$$

avec T le temps de la mesure, (t_1, \dots, t_k) les instants d'arrivées sur l'intervalle $(0, T]$ et k le nombre de ces instants d'arrivées.

On suppose que l'intensité du processus est constante sur chaque heure. C'est ce qui permet de simplifier la log-vraisemblance (voir [Baddeley et al., 2016]) :

$$l(\theta|t_1, \dots, t_k, T) = \sum_{i=1}^k \ln \lambda(t_i; \theta) - \sum_{j=1}^T \lambda(j; \theta) = \sum_{j=1}^T (y_j \ln \lambda(j; \theta) - \lambda(j; \theta)) \quad (36)$$

avec $y_j = 1$ pour t_j avec libération de spores et $y_j = 0$ pour t_j sans libération de spores.

L'équation (36) s'écrit :

$$l(\theta|t_1, \dots, t_k, T) = \sum_{j=1}^T (y_j \theta^\top Z(j) - \exp(\theta^\top Z(j))). \quad (37)$$

On peut montrer que l'équation (37) est égale à la log vraisemblance d'un modèle de régression avec la réponse $[y_1, \dots, y_T]$ et les covariables météorologiques. Pour ajuster ce modèle, on peut utiliser des logiciels statistiques qui permettent d'ajuster des modèles linéaires généralisés, par exemple la fonction "glm" de **R**. Le Tableau 4 donne les estimateurs du maximum de vraisemblance du vecteur θ .

Pour faire de la sélection de variable, on considère l'hypothèse : "la variable Z_i n'influence pas l'intensité du processus" :

$$H_0 : \theta_i = 0.$$

L'estimateur du maximum de vraisemblance est asymptotiquement normalement distribué, ainsi sous hypothèse nulle :

$$z = \frac{\hat{\theta}_i}{sd(\hat{\theta}_i)} \sim \mathcal{N}(0, 1).$$

avec i la i ème covariable et $sd(\hat{\theta}_i)$ l'écart type de l'estimateur.

Pour tester H_0 au niveau α , on calcule la valeur de z et on la compare avec la valeur critique de test z_α . Si $|z| > z_\alpha$ alors on rejette H_0 .

Le Tableau 4 donne les estimateurs et les valeurs de test des hypothèses H_0 .

	Estimate	Std. Error	z value	Pr(> z)	Niveau du test
Constante	-7.4	0.57	-12	< 2e-16	***
LEAFWET	1.56	0.14	11	< 2e-16	***
RELHUMID	0.04	0.005	7.6	2.54e-14	***
BRIGHTNESS	0.95	0.13	7.2	5.32e-13	***
PRECIPIT	0.05	0.05	1.0	0.31430	
TEMP1	0.01	0.012	0.94	0.34710	
SOLAR	0.0004	0.0004	1.15	0.24889	
ANNEE2011	0.31	0.1	3.0	0.00243	**

TABLE 4 – Estimations des paramètres θ_i au modèle (deuxième colonne), écart-type (troisième colonne), staitistique de test H_0 quatrième colonne, p -value cinquième colonne et niveau du test (dernière colonne). Les trois étoiles '***' désignent on rejette H_0 au niveau $\alpha = 0.1\%$ et '**' on rejette H_0 au niveau 1% .

Conformément à ce résultat, on garde le modèle avec les variables : "humidité des feuilles", "humidité relative de l'air", "luminosité" et "effet de l'année". Ces variables ont donc de l'influence sur l'intensité. Pour avoir une idée de cette influence, il faut se référer aux paramètres. Dans le cas d'une variable qualitative comme l'humidité des feuilles, la présence d'humidité foliaire augmente $\log(\lambda(t))$ de 1.56. Lorsque que les variables sont quantitatives, on interprète les paramètres comme l'augmentation de la log – intensité pour un incrément de 1 de ces variables. Pour l'humidité relative de l'air, une hausse de 1% d'humidité relative augmente $\log(\lambda(t))$ de 0.04.

La Figure 13 montre un graphique correspondant au nombre d'instant de libération de spores par jour ainsi que le nombre d'instant prédits par le processus de Poisson non homogène. On en déduit que le processus ajusté présente le même comportement que les données, à savoir des prédictions de libérations de spores semblables aux données réelles. Néanmoins on constate également que le nombre d'instant de libération prédits diffère du nombre exact.

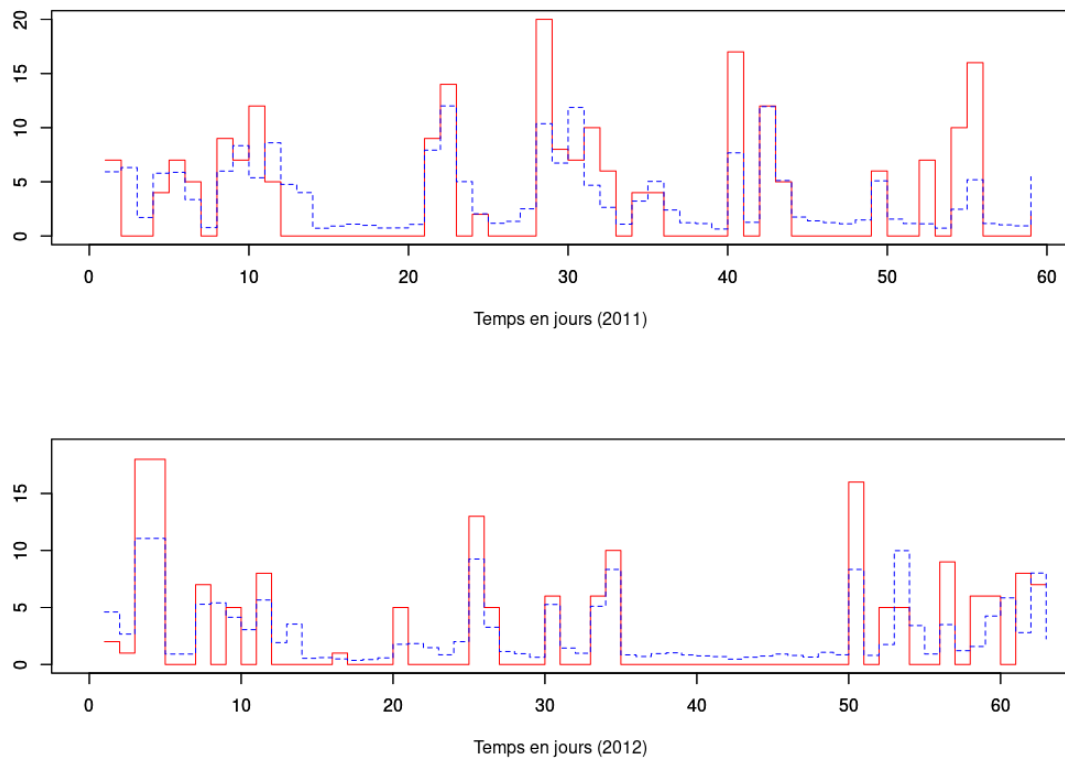


FIGURE 13 – Nombre d’instant de libération de spores observés par jour (ligne continue rouge) et nombre d’instant de libération de spores prédits par le processus de Poisson avec une intensité log-linéaire (ligne discontinue bleue) en 2011 (a) et en 2012 (b).

6.3 Ajustement du processus de Hawkes

L'intensité conditionnelle du processus de Hawkes est donné par l'équation (28) :

$$\lambda^*(t) = \lambda + \sum_{t_i < t} \alpha \exp(-\beta t)$$

avec λ, α et β des réels positifs tel que $\alpha < \beta$.

Dans ce cas la log-vraisemblance s'écrit :

$$l(\theta|t_1, \dots, t_k) = \sum_{i=1}^k \ln(\lambda + \alpha A(i)) - \lambda t_k + \frac{\alpha}{\beta} \sum_{i=1}^k \left[e^{-\beta(t_k - t_i)} - 1 \right]$$

avec $\theta = (\lambda, \alpha, \beta)$, t_i les instants de libérations de spores et k leur nombre. On traite les deux années séparément car l'équation 32 ne permet pas d'inclure un effet de l'année.

Les résultats de l'estimation des paramètres sont présentés dans le Tableau 5. Les intervalles de confiance asymptotique à 95% des paramètres ne contiennent pas 0, on en déduit que la fonction $\mu(\cdot)$ n'est pas nulle et que le modèle de Hawkes est significativement meilleur que le modèle de Poisson homogène. De plus, les valeurs des paramètres impliquent une grande contribution de la fonction d'excitation $\mu(\cdot)$ et donc des événements eux mêmes. En effet à l'état stationnaire $\mathbb{E}[\lambda^*(t)] = \frac{\lambda}{1-n} = 0.16 >> \lambda = 0.027$ pour l'année 2011.

	λ	$Confint(\lambda)$	α	$Confint(\alpha)$	β	$Confint(\beta)$
Année 2011	0.02073	[0.0205,0.0209]	0.2639	0.2625,0.2652	0.3028	0.3014,0.3041]
Année 2012	0.01692	[0.01678,0.01706]	0.3900	0.3870,0.3910	0.4487	0.4467,0.4506]

TABLE 5 – Estimations des paramètres du processus de Hawkes pour les données épidémiques en 2011 et 2012.

6.4 Comparaisons des modèles

Sélection des modèles critère Akaike Dans cette sous-section on utilise des méthode pour valider les modèles existants : le critère d'Akaike, la comparaison avec des courbes simulées et l'analyse résiduelle sont étudiés.

le AIC est égal à :

$$AIC = 2p - 2\log(L)$$

où p est le nombre de paramètres et $\log(L)$ la log-vraisemblance. Selon le critère d'Akaike, les modèles sont classés en fonction de leur AIC et on choisit le modèle qui a l'indice le plus faible. La comparaisons de l'indice AIC pour les trois processus, Poisson homogène, Poisson non homogène et Hawkes, montre que le processus de Hawkes est le meilleur modèle avec un indice légèrement meilleur que le processus de Poisson non homogène.

Simulation des processus ajustés Un outil graphique consiste à tracer des courbes simulés selon plusieurs modèles. Après avoir estimés les paramètres des différents processus (Poisson homogène, Poisson non homogène et Hawkes) on simule plusieurs trajectoires que l'on compare avec nos données de libération de spores (voir la colonne de gauche de la Figure 14). La comparaison

AIC	Année 2011	Année 2012
Poisson homogène	1242.53	1102.40
Poisson non homogène	983.22	751.78
Hawkes	917.29	726.95

TABLE 6 – Comparaison des AIC pour le modèle de Poisson homogène, le modèle de Poisson non homogène avec une intensité log-linéaire dépendant de trois covariables et le modèle de Hawkes avec une fonction d'excitation exponentielle.

avec le processus de Poisson homogène montre que la diffusion des sauts du processus s'effectue uniformément et non pas par paquet comme pour les données. Les courbes simulées selon le processus de Poisson non homogène ont le même comportement que la courbe exacte, à savoir des sauts successifs et des larges périodes sans sauts. Néanmoins la courbe exacte se trouve au bord des courbes simulées ce qui montre que la courbe des données a peu de chance d'appartenir au processus de Poisson non homogène. Le troisième graphique des courbes simulées témoigne de la grande variance du processus de Hawkes. La courbe des données se trouve au centre des courbes simulées et leur comportement se ressemble.

D'après ce diagnostic graphique, deux processus se distinguent : le processus de Poisson non homogène et le processus de Hawkes. Les courbes simulées reproduisent la dynamique des données et peut être utilisé pour prédire d'autres données réelles.

Analyse résiduelle L'étude des résidus des processus est utilisé pour comparer les modèles. On rappelle que si on a une réalisation d'un processus ponctuel (t_1, \dots, t_k) avec un compensateur $\Lambda(\cdot)$, alors les résidus définis par $R(i) = \Lambda(t_i) - \Lambda(t_{i-1})$ (avec $t_0 = 0$) forment un échantillon de la loi exponentielle de paramètre 1. Pour vérifier cela, les quantiles de la loi exponentielle de paramètre 1 sont comparés aux quantiles des résidus à l'aide d'un QQ-plot.

Sur la colonne de droite de la Figure 14, trois listes de résidus sont générées. Une à partir des paramètres ajustés du processus de Poisson homogène, la seconde avec les paramètres du processus de Poisson non homogène et le dernier avec les paramètres du processus de Hawkes. Pour les deux premiers graphiques, on constate que les points ne sont pas placés le long de la première bissectrice et que l'étendue des quantiles est largement supérieur aux quantiles théoriques. Ce qui permet d'exclure que ces processus modélisent bien les données. Le graphique des quantiles du processus de Hawkes est plus favorable. L'étendue des quantiles est identique, et les points sont davantage situés autour de la première bissectrice. On en déduit que le processus de Hawkes est le meilleur des trois modèles après le modèle de Poisson non homogène.

Néanmoins cela ne signifie pas que c'est un bon modèle. En effet le test de Kolmogorov-Smirnoff retourne une p-valeur de 10^{-5} . Ce qui signifie que les résidus du processus de Hawkes sont significativement différents d'une loi exponentielle de paramètre 1, et donc que les données ne sont probablement pas issues d'un processus de Hawkes stationnaire avec une fonction d'excitation exponentielle.

7 Conclusion

Dans ce stage, j'ai étudié les processus ponctuels et en particulier le processus de Poisson homogène, non homogène et le processus de Hawkes. J'ai analysé les propriétés des processus et réalisé

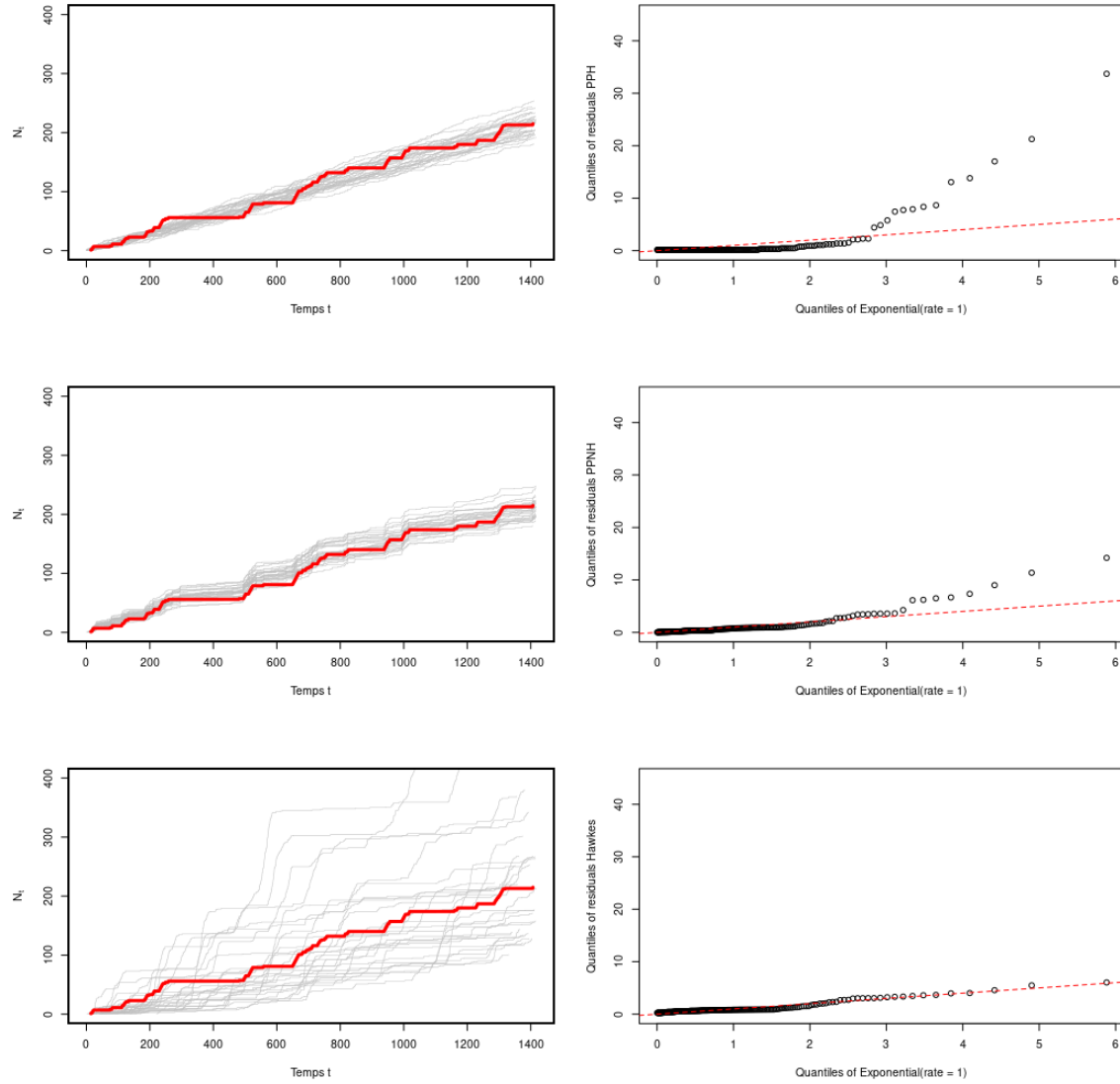


FIGURE 14 – Les données de libération de spores de l'année 2011 sous la forme d'un processus de comptage (en rouge) et les simulations selon le modèle de Poisson homogène, Poisson non homogène et Hawkes. Colonne de droite : QQ-plot des résidus et validation du modèle.

leurs démonstrations mathématiques. J’ai ensuite étudié et implémenté les algorithmes de simulation des différents processus. Pour le processus de Poisson non homogène, j’ai analysé l’algorithme d’amincissement de Lewis-Shedler et l’algorithme de l’inverse du compensateur. Pour le processus de Hawkes, je me suis intéressé à l’algorithme d’amincissement modifié d’Ogata et l’algorithme de l’inverse du compensateur. J’ai utilisé la méthode du maximum de vraisemblance pour estimer les paramètres des différents modèles et j’ai étudié les propriétés asymptotiques de l’estimateur du maximum de vraisemblance. J’ai utilisé l’analyse résiduelle pour valider les modèles.

Dans le contexte de la tavelure des pommiers, un grand intérêt est porté à la libération de spores du champignon *Venturia Inaequalis*. Dans l’optique de modéliser la propagation des spores, on ajuste les modèles aux données de libération de spores. Trois processus sont ajustés : le processus de Poisson homogène, le processus de Poisson non homogène avec une intensité log –linéaire permettant d’inclure des covariables climatiques et le processus de Hawkes avec une intensité d’arrière plan constante et une fonction d’excitation exponentielle. Le processus de Poisson homogène ne reproduit pas la dynamique des données à cause de la diffusion uniforme des points et des données groupées. L’ajustement au processus de Poisson non homogène montre une dépendance entre les instants de libération de spores et l’humidité des feuilles, l’humidité relative de l’air et la luminosité. Avec l’inclusion de ces covariables dans le modèle, on constate que les prédictions sont fidèles car elles conservent la dynamique des données avec un phénomène d’agrégation. Néanmoins ce modèle ne prend pas en compte la dépendance des événements. L’ajustement au processus de Hawkes montre que le caractère auto-excitant du processus modélise bien les données clusterisées de libération de spores. Le facteur de branchement élevé est significatif d’une forte dépendance entre les événements. Seulement en l’absence d’une intensité d’arrière plan dépendant des conditions météorologiques, on ne peut pas l’utiliser pour prévoir de futures libérations de spores. D’après la comparaison des modèles, le processus de Hawkes est celui qui s’ajuste le mieux aux données.

Dans la perspective d’améliorer le modèle, il est souhaitable d’étudier le processus de Hawkes dans le cas non stationnaire ce qui permettrait de relier dans le même modèle, le caractère auto-excitant des instants d’arrivées et la contribution des données climatiques.

Ce stage m’a permis de développer mes connaissances en probabilité, statistique et modélisation stochastique. J’ai approfondie mes compétences dans le langage R avec lequel j’ai implémenté des programmes, visualisé et modélisé des données.

Références

- [Allal, 2020] Allal, L. B. (2020). Processus de hawkes. *Ecole des Mines de Nancy*.
- [Baddeley et al., 2016] Baddeley, A., Rubak, E., and Turner, R. (2016). *Spatial point patterns : methodology and applications with R*. Chapman Hall/CRC Interdiscip. Stat. Ser. Boca Raton, FL : CRC Press.
- [Cadre, 2022] Cadre, B. (2022). *Processus stochastique M2 Rennes*. https://w3.ens-rennes.fr/math/people/benoit.cadre/fichiers/LIVRE_PROC.pdf.
- [Da Fonseca and Zaatour, 2014] Da Fonseca, J. and Zaatour, R. (2014). Hawkes process : Fast calibration, application to trade clustering, and diffusive limit. *Journal of Futures Markets*, 34(6) :548–579.
- [Daley and Vere-Jones, 2003] Daley, D. J. and Vere-Jones, D. (2003). *An introduction to the theory of point processes. Vol. I : Elementary theory and methods*. Probab. Appl. New York, NY : Springer, 2nd ed. edition.

- [Decreusefond and Moyal, 2023] Decreusefond, L. and Moyal, P. (2023). Réseaux aléatoires. <https://decrease.github.io/site>.
- [Hawkes, 1971] Hawkes, A. G. (1971). Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society Series B : Statistical Methodology*, 33(3) :438–443.
- [INRAE, 2016] INRAE (2016). *Venturia inaequalis* (cooke) g. winter, (1875). <http://ephytia.inra.fr/fr/C/20846/Pomme-Venturia-inaequalis-tavelure-du-pommier>.
- [Jankowski and Masny, 2020] Jankowski, P. and Masny, S. (2020). Influence of moisture on maturation rate of the *Venturia inaequalis* (Cooke) Wint. ascospores in central Poland. *Journal of Plant Diseases and Protection*, 127(2) :155–163.
- [Laub et al., 2015] Laub, P. J., Taimre, T., and Pollett, P. K. (2015). Hawkes processes. *arXiv preprint arXiv :1507.02822*.
- [Lewis and Shedler, 1979] Lewis, P. A. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Nav. Res. Logist. Q.*, 26 :403–413.
- [M.Deaconu, 2022] M.Deaconu (2022). *Cours de Master 2, Modélisation stochastique, UNIVERSITÉ DE LORRAINE, POLYCOPIÉ*.
- [NOR, 2019] NOR (12 octobre 2019). Décret n 2019-1046 du 10 octobre 2019 relatif à l’organisation et au fonctionnement de l’institut national de recherche pour l’agriculture, l’alimentation et l’environnement. *JORF*. <https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000039207211>.
- [Ogata, 1978] Ogata, Y. (1978). The asymptotic behaviour of maximum likelihood estimators for stationary point processes. *Ann. Inst. Stat. Math.*, 30 :243–261.
- [Ogata, 1981] Ogata, Y. (1981). On Lewis’ simulation method for point processes. *IEEE Trans. Inf. Theory*, 27 :23–31.
- [Ogata, 1988] Ogata, Y. (1988). Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*, 83(401) :9–27.
- [Omi et al., 2017] Omi, T., Hirata, Y., and Aihara, K. (2017). Hawkes process model with a time-dependent background rate and its application to high-frequency financial data. *Physical Review E*, 96(1) :012303.
- [Ozaki, 1979] Ozaki, T. (1979). Maximum likelihood estimation of Hawkes’ self-exciting point processes. *Ann. Inst. Stat. Math.*, 31 :145–155.
- [Rancane et al., 2007] Rancane, R., Eihe, M., and Jankovska, L. (2007). Adaption of simulation model rimpro for primary apple scab control in latvia. In *VIII International Symposium on Modelling in Fruit Research and Orchard Management 803*, pages 69–76.
- [Rasmussen, 2011] Rasmussen, J. G. (2011). Temporal point processes : the conditional intensity function. *Lecture Notes, Jan*.
- [Rathbun and Cressie, 1994] Rathbun, S. L. and Cressie, N. (1994). Asymptotic properties of estimators for the parameters of spatial inhomogeneous poisson point processes. *Advances in Applied Probability*, 26(1) :122–154.
- [Ruwet, 2007] Ruwet, C. (2007). Processus de Poisson. *Université de Liège*.
- [Schoenberg et al., 2019] Schoenberg, F. P., Hoffmann, M., and Harrigan, R. J. (2019). A recursive point process model for infectious diseases. *Ann. Inst. Stat. Math.*, 71(5) :1271–1287.

8 Annexe

Dans cette section, les codes **R** des algorithmes de simulation et des procédures d'inférence sont présentés.

8.1 Simulation du Processus de Poisson homogène

Code R 8.1. *Simulation d'un processus de Poisson avec la répartition uniforme (voir 1).*

```
#DESCRIPTION Génère un processus de Poisson homogène en utilisant la
#répartition uniforme des points.
#Entrée: Une intensité lambda et une limite de temps T.
#Sortie: Une liste de points répartis sur (0,T].
simu_PPH<-function(lambda,T){
  N<-rpois(1,lambda*T)
  Liste<-c(sort(runif(N,0,T)))
  return(Liste)
}
simu_PPH(2,20)
```

Code R 8.2. *Simulation d'un processus de Poisson par renouvellement (voir 2).*

```
#Description: Génère un processus de Poisson homogène avec un
#processus de renouvellement.
#Entrée      : le paramètre lambda et une borne de temps T.
#Sortie      : Une liste de points ordonnés sur (0,T].
simu_PPH2<-function(lambda,T){
  Liste<-c()
  t<-0
  while(t<T){
    t<-t+rexp(1,rate=lambda)
    Liste<-c(Liste,t)
  }
  Liste<-Liste[-length(Liste)]
  return(Liste)
}
simu_PPH2(2,20)
```

8.2 Simulation du processus de Poisson non homogène

Code R 8.3. *Simulation d'un processus de Poisson non homogène par l'inverse du compensateur lorsque l'inverse est connue (voir 3).*

```
#Description: Simule un processus de Poisson non homogène lorsque l'inverse
#du compensateur est connue.
#Entrée      : Le compensateur, son inverse et une borne de temps T.
#Sortie      : Une liste de points ordonnés sur (0,T].
simuPPNH_inversion_simple<-function(compens,inverse_compens,T){
```

```

T_aux<-compens(T)
Liste<-simu_PPH(1,T_aux)
for (i in 1:length(Liste)){
  Liste[i]<-inverse_compens(Liste[i])
}
return(Liste)
}

#Le compensateur de l'intensité linéaire lambda(t)=t.
compens<-function(t){
  return((t**2)/2)
}
#L'inverse de ce compensateur.
inverse_compens<-function(t){
  return(sqrt(2*t))
}
T<-15
Liste<-simuPPNH_inversion_simple(compens,inverse_compens,T)

```

Code R 8.4. *Simulation d'un processus de Poisson non homogène par l'inverse du compensateur de manière itérative (voir 4).*

```

#Description: Simule un processus de Poisson non homogène par l'inverse du
#compensateur de manière itérative.
#Entrée      : La fonction d'intensité: intensite
#            : Le compensateur: compensateurPPNH
#            : Une borne de temps T.
#Sortie      : Une liste de points ordonnés sur (0,T).
simu_PPNH_inversion<-function(intensite,compensateurPPNH,T){
  t_k<-0
  Liste<-c()
  while(t_k<T){
    U<-runif(1,0,1)
    fonction_root<-function(t){# La fonction à annuler
      return(compensateurPPNH(t)-compensateurPPNH(t_k)+log(U))
    }
    if(fonction_root(T)<0){#Si fonction_root(T)<0, alors le point simulé se trouve
      #plus loin que la borne T. On retourne alors Liste.
      return(Liste)
    }
    t_k<-uniroot(fonction_root,interval=c(t_k,T),tol=10**(-7))$root
    #uniroot trouve la racine de uniroot avec une tolerance à 10**(-7).
    Liste<-c(Liste,t_k)
  }
}

```

#Description: Une fonction d'intensité périodique utilisée pour la simulation.

```

#Entrée :t
#Sortie: sinfun(t)
sinfun<-function(t){
  y<-4+4*sin(1/3*t)
  y
}
#Description: La fonction compensateur de la fonction ci-dessus sinfun.
#Entrée      : t
#Sortie      : Un scalaire.
compensateurPPNH<-function(t){
  return(4*t-12*cos(t/3)+12)
}

T<-40
Liste<-simu_PPNH_inversion(sinfun,compensateurPPNH,T)

```

Code R 8.5. *Simulation d'un processus de Poisson non homogène par amincissement (voir 5).*

```

#Description: Génère une liste de points suivant un processus de Poisson non
#homogène avec une procédure d'amincissement.
#Entrée:      Une fonction bornée:fun
#              Un majorant de cette fonction max_funT
#              Une borne de temps T.
#Sortie:      Une liste de points ordonnés sur (0,T].
simu_PPNH_thinning<-function(fun,max_funT,T)
{
  N<-rpois(1,T*max_funT)
  Liste_aux<-sort(runif(N,0,T))#Simulation d'un processus de Poisson homogène
  #d'intensité max_funT sur l'intervalle de temps (0,T].
  Liste<-c()
  for (i in Liste_aux){
    if (runif(1,0,1)<fun(i)/max_funT){
      Liste<-c(Liste,i)
    }
  }
  Liste
}
Liste<-simu_PPNH_thinning(sinfun,8,30)

```

8.3 Validation du modèle : Processus de Poisson non homogène

Code R 8.6. *Validation du modèle pour un processus de Poisson non homogène (voir 4.4).*

```

#Description: Calcule la Liste des résidus d'un processus de Poisson
#non homogène d'intensité lambda(.).
#Entrée      : Une liste de points ordonnés
#              : La fonction compensateur.

```

```

#Sortie      : La liste des résidus.
residusPPNH_fun<-function(Liste,compensateurPPNH){
Liste_aux<-c()
for (i in Liste){
  Liste_aux<-c(Liste_aux,compensateurPPNH(i))
}
residus<-c(Liste_aux[1],diff(Liste_aux))
return(residus)
}
T<-80
Liste<-simu_PPNH_inversion(sinfun,compensateurPPNH,T)
residusPPNH<-residusPPNH_fun(Liste,compensateurPPNH)
ks.test(residusPPNH,pexp,1)

```

8.4 Simulation du processus de Hawkes

Code R 8.7. *Simulation d'un processus de Hawkes par l'inverse du compensateur (voir 7).*

```

#Description : Simule le processus de Hawkes par l'inverse du compensateur.
#Entrée      : Les paramètres lambda, alpha et beta, et une borne de temps T.
#Sortie      : Une liste de points ordonnés sur l'intervalle (0,T].
simu_hawkes_inversion<-function(lambda,alpha,beta,T){
  t_k<- -log(runif(1,0,1))/lambda
  k<-1
  n_ratio<-alpha/beta
  Liste_Hawkes<-c(t_k)
  S_k<-1
  while(t_k<T){
    U<-runif(1,0,1)
    fonction_root<-function(t){ #La fonction à annuler
      return(log(U)+lambda*(t-t_k)-n_ratio*S_k*(exp(-beta*(t-t_k))-1))
    }
    if (fonction_root(T)<0){ #Si la fonction n'est pas supérieur à 0 au point T
      return(Liste_Hawkes) #on retourne la Liste. Cela signifie que le point recherché
        #est plus loin que la borne T.
    }
    t_k<-uniroot(fonction_root,interval=c(0,T),tol=10**(-7))$root #La racine trouvée avec uniroot
    #avec une tolérance à 10**(-7).
    Liste_Hawkes<-c(Liste_Hawkes,t_k)
    k<-k+1
    S_k<-1+S_k*exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[k-1]))
  }
}

lambda<-0.5
alpha<-1
beta<-1.5

```



```
T<-50
Liste_Hawkes<-simu_hawkes_inversion(lambda,alpha,beta,T)
```

Code R 8.8. *Simulation d'un processus de Hawkes par l'algorithme d'amincissement modifié d'Ogata (voir 6).*

```
#La fonction ci-dessous est une fonction auxiliaire utilisée dans
#la procédure d'amincissement
#Description: Calcul la valeur de l'intensité conditionnelle d'un
#processus de Hawkes au temps t.
#Entree:      Le temps t.
#             : Les paramètres lambda, alpha et beta.
#             : La liste des points du processus Liste_Hawkes.
#Sortie       : La valeur de l'intensité conditionnelle.
lambda_star_fun<-function(t,lambda,alpha,beta,Liste_Hawkes)
{N<-length(Liste_Hawkes[Liste_Hawkes<=t])
res<-lambda
if (N!=0){
  for (i in 1:N){
    res<-res+alpha*exp(-beta*(t-Liste_Hawkes[i]))
  }
}
return(res)
}

#Description: Fonction simulant un processus de Hawkes avec l'algorithme
#d'amincissement d'Ogata.
#Entrée       : Les paramètres du modèle lambda ,alpha et beta
#              : Une borne de temps: t.
#Sortie       : Une liste de points ordonnés sur (0,T].
hawkes_thinning<-function(lambda,alpha,beta,T){
  t<-0
  Liste_Hawkes<-c()
  M<-lambda
  while (t<T){
    W<-rexp(1,rate=M)
    U<-runif(1,0,M)
    if (U<=lambda_star_fun(t+W,lambda,alpha,beta,Liste_Hawkes)){
      t<-t+W
      M<-alpha+lambda_star_fun(t+W,lambda,alpha,beta,Liste_Hawkes)
      Liste_Hawkes<-c(Liste_Hawkes,t)
    }
    else if(U>lambda_star_fun(t+W,lambda,alpha,beta,Liste_Hawkes)){
      t<-t+W
      M<-lambda_star_fun(t+W,lambda,alpha,beta,Liste_Hawkes)
    }
  }
}
```

```

    }
    if (length(Liste_Hawkes) > T) {
      Liste_Hawkes <- Liste_Hawkes[-length(Liste_Hawkes)]
    }
    if (length(Liste_Hawkes) == 1 && length(Liste_Hawkes) > T) {
      Liste_Hawkes <- c()
    }
    return(Liste_Hawkes)
  }

lambda=0.5
alpha=1
beta=1.5
T=50
Liste_Hawkes <- hawkes_thinning(lambda, alpha, beta, T)

```

8.5 Recherche du maximum de vraisemblance Hawkes

Code R 8.9. Recherche du maximum de vraisemblance d'un processus de Hawkes donnée en (5.4).

Les listes *A_liste* et *B_liste* sont calculées de manière récursive puis les codes des dérivés de la fonction de vraisemblance sont proposés. Enfin la fonction log-vraisemblance et la fonction du gradient de cette log-vraisemblance sont implémentées puis le maximum est calculé avec l'outil *optim* de R.

```

#Description : Calcul le maximum de la log-vraisemblance
#des paramètres du processus de Hawkes
#Entrée      : Une liste de points ordonnées Liste
#            : La position initiale (Choisir NULL pour obtenir la valeur
#(lambda, poissonhomogene, 1, 1) par défaut.
#Sortie      : le vecteur des paramètres (lambda, alpha, beta).
#            : Le maximum de la log-vraisemblance Log_lik
#            : Le gradient au point trouvé
#            : La matrice hessienne empirique
mle_Hawkes_fun <- function(Liste, init) {

  k <- length(Liste)
  lambda <- length(Liste) / Liste[length(Liste)]
  if (is.null(init)) {
    init <- c(lambda, 1, 1)
  }
  lnL <- function(x) {
    return(log_likelihood(x, Liste, k))
  }
  grdL <- function(x) {
    return(gradient_likelihood(x, Liste_Hawkes, k))
  }
  borne_inf <- c(10**(-10), 10**(-10), 10**(-10))

```

```

Opt_mle<-optim(init, lnL,grdL,
               method = "L-BFGS-B",lower=borne_inf,control=list(fnscale=-1),hessian=TRUE)
param<-Opt_mle$par
hessianLog=Opt_mle$hessian
return(list(param=Opt_mle$par,Log_lik=lnL(param)
            ,gradient=grdL(param),Matrice_hessienne=hessianLog))
}

```

```

Liste_Hawkes<-simu_hawkes_inversion(0.5,1,1.5,100)
mle_hwakes_fun(Liste_Hawkes,NULL)

```

```

#Description: Fonction retournant la valeur de la log-vraisemblance.
#Entrée      : un vecteur x=(lambda,alpha,beta).
#Sortie      : Une valeur numérique qui est l'image de x par log(L).
log_likelihood<-function(x,Liste_Hawkes,k){
  lambda<-x[1]
  alpha<-x[2]
  beta<-x[3]
  n_ratio<-alpha/beta
  k<-length(Liste_Hawkes)
  A_liste<-A_liste_fun(Liste_Hawkes,beta,k)
  res<-0
  for (i in 1:k){
    res<-res+log(lambda+alpha*A_liste[i])
  }
  for (j in 1:k){
    res<-res+n_ratio*(exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[j]))-1)
  }
  res<-res-lambda*Liste_Hawkes[k]
  res
}

```

```

#Description: Fonction calculant le gradient de la vraisemblance.
#Entrée      : Un vecteur x=(lambda,alpha,beta).
#Sortie      : Un vecteur du gradient de x.
gradient_likelihood<-function(x,Liste_Hawkes){
  lambda<-x[1]
  alpha<-x[2]
  beta<-x[3]
  k<-length(Liste_Hawkes)
  return(c(derive_lambda(lambda,alpha,beta,Liste_Hawkes,k)
            ,derive_alpha(lambda,alpha,beta,Liste_Hawkes,k)

```

```

    ,derive_beta(lambda,alpha,beta,Liste_Hawkes,k)))
}

###Étude de la vraisemblance et du maximum de vraisemblance.
#Description: Calcule les sommes A_i de façon récursive.
#Entrée:      La liste de points Liste_Hawkes
#             Le paramètre beta
#             Le nombre de points k.
#Sortie:      La liste de tout les A_i.
A_liste_fun<-function(Liste_Hawkes,beta,k)
{
  A_liste<-c(0)
  for (i in 2:k){
    A_liste<-c(A_liste,exp(-beta*(Liste_Hawkes[i]-Liste_Hawkes[i-1]))*
                (1+A_liste[i-1])))
  }
  A_liste
}

#Description :Calcul les B_i de façon récursive.
#Entrée      :Le paramètre beta, A_liste et k.
#Sortie      :Une liste B_liste contenant tout les B_i.
B_liste_fun<-function(Liste_Hawkes,beta,A_liste,k)
{B_liste<-c(0)
for (i in 2:k){
  B_liste<-c(B_liste,exp(-beta*(Liste_Hawkes[i]-Liste_Hawkes[i-1]))*
              ((Liste_Hawkes[i]-Liste_Hawkes[i-1])*(A_liste[i-1]+1)+B_liste[i-1])))
}
B_liste
}

#Description: Calcul la dérivée de la fonction de vraisemblance
#par rapport à lambda.
#Entrée      : Les paramètres: lambda, alpha et beta.
#Sortie      : Un scalaire.
derive_lambda<-function(lambda,alpha,beta,Liste_Hawkes,k){
  res<-0
  A_liste<-A_liste_fun(Liste_Hawkes,beta,k)
  for (i in 1:k){
    res<-res+1/(lambda+alpha*A_liste[i])
  }
  res<-res-Liste_Hawkes[k]
  return(res)
}

```

```

#Description: Calcul la dérivée de la fonction de vraisemblance
#par rapport à alpha.
#Entrée: Les paramètres: lambda, alpha et beta.
#Sortie: Un scalaire.
derive_alpha<-function(lambda,alpha,beta,Liste_Hawkes,k){
  res<-0
  A_liste<-A_liste_fun(Liste_Hawkes,beta,k)
  for (i in 1:k){
    res<-res+ A_liste[i]/(lambda+alpha*A_liste[i])
    +(1/beta)*(exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))-1)
  }
  return(res)
}

```

```

#Description: Calcul la dérivée de la fonction de vraisemblance
#par rapport à beta.
#Entrée: Les paramètres: lambda, alpha et beta.
#Sortie: Un scalaire.
derive_beta<-function(lambda,alpha,beta,Liste_Hawkes,k){
  res<-0
  A_liste<-A_liste_fun(Liste_Hawkes,beta,k)
  B_liste<-B_liste_fun(Liste_Hawkes,beta,A_liste,k)
  for (i in 1:k){
    res<-res-(alpha*B_liste[i])/(lambda+alpha*A_liste[i])-(alpha/beta**2)
    *(exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))-1)-(alpha/beta)
    *(Liste_Hawkes[k]-Liste_Hawkes[i])*exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))
  }
  return(res)
}

```

8.6 Hessienne et normalité asymptotique Hawkes

Code R 8.10. *Calcul de la matrice Hessienne de la log-vraisemblance d'un processus de Hawkes et calcul des intervalles de confiances des paramètres (voir l'équation (33)).*

#Les codes suivant permettent de trouver la matrice Hessienne.
 #On s'intéresse ensuite à la normalité asymptotique.

```

#Description: Calcule les sommes C_i de manière récursive.
#Entrée      : La liste de Points Liste_Hawkes
#            : Le paramètre beta
#            : La liste A_liste
#            : La liste B_liste

```

```

#           : Le nombre de points k.
#Sortie     : La liste C_liste qui contient les C_i.
C_liste_fun<-function(Liste_Hawkes,beta,A_liste,B_liste,k)
{
  C_liste<-c(0)
  for (i in 2:k){
    aux<-exp(-beta*(Liste_Hawkes[i]-Liste_Hawkes[i-1]))*((Liste_Hawkes[i]
    -Liste_Hawkes[i-1])**2+C_liste[i-1]+A_liste[i-1]*((Liste_Hawkes[i]
    -Liste_Hawkes[i-1])**2)+2*B_liste[i-1]*(Liste_Hawkes[i]-Liste_Hawkes[i-1]))
    C_liste<-c(C_liste,aux)
  }
  C_liste
}

#Les fonctions suivantes cacclulent les dérivés secondes de la log-vraisemblance.
#Description: Calcule la dérivé seconde lambda carré de la log-vraisemblance.
#Entrée      : Les paramètres lambda, alpha et beta
#            : La liste des sommes A_i
#            : La liste des points Liste_Hawkes
#            : Le nombre de points k.
#Sortie      : Un scalaire.
derive_lambda_carre<-function(lambda,alpha,beta,A_liste,Liste_Hawkes,k){
  res=0
  for (i in 1:k){
    res<- res -1/((lambda+alpha*A_liste[i])**2)
  }
  res
}

#Les cinq fonctions suivantes calculent le reste des dérivés secondes.
#Pour certaines dérivés les sommes B_i et C_i sont requises.
derive_lambda_alpha<-function(lambda,alpha,beta,A_liste,Liste_Hawkes,k){
  res<-0
  for (i in 1:k){
    res<- res - A_liste[i]/((lambda+alpha*A_liste[i])**2)
  }
  res
}

derive_lambda_beta<-function(lambda,alpha,beta,A_liste,B_liste,Liste_Hawkes,k){
  res<-0
  for (i in 1:k){
    res<- res + alpha*B_liste[i]/((lambda+alpha*A_liste[i])**2)
  }
  res
}

```

```

derive_alpha_carre<-function(lambda,alpha,beta,A_liste,Liste_Hawkes,k){
  res<-0
  for (i in 1:k){
    res<- res - (A_liste[i]/(lambda+alpha*A_liste[i]))**2
  }
  res
}

derive_alpha_beta<-function(lambda,alpha,beta,A_liste,B_liste,Liste_Hawkes,k){
  res<-0
  for (i in 1:k){
    res<-res+ alpha*A_liste[i]*B_liste[i]/((lambda+alpha*A_liste[i])**2)
    - B_liste[i]/(lambda+alpha*A_liste[i])- (1/beta)*(Liste_Hawkes[k]
    -Liste_Hawkes[i])*exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))
    -(1/beta**2)*(exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))-1)
  }
  res
}

derive_beta_carre<-function(lambda,alpha,beta,A_liste,B_liste,C_liste
                           ,Liste_Hawkes,k){
  res<-0
  for (i in 1:k){
    res<-res+alpha*((1/beta)*((Liste_Hawkes[k]-Liste_Hawkes[i])**2)
    *exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))+(2/beta**2)*(Liste_Hawkes[k]
    -Liste_Hawkes[i])*exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))+(2/beta**3)
    *(exp(-beta*(Liste_Hawkes[k]-Liste_Hawkes[i]))-1))
    res<-res+ alpha*C_liste[i]/(lambda+alpha*A_liste[i])
    -(alpha*B_liste[i]/(lambda+alpha*A_liste[i]))**2
  }
  res
}

#####
#####Fonction Hessienne#####

#Description: Fonction retournant la matrice Hessienne théorique.
#Entrée      : La liste de points Liste_Hawkes
#            : Les paramètres lambda, alpha et beta
#            : Le nombre de points k.
#Sortie      : La matrice Hessienne sous la forme d'un tableau 3*3.
hessienne_fun<-function(Liste_Hawkes,lambda,alpha,beta){

```

```

A_liste<-A_liste_fun(Liste_Hawkes,beta,k)
B_liste<-B_liste_fun(Liste_Hawkes,beta,A_liste,k)
C_liste<-C_liste_fun(Liste_Hawkes,beta,A_liste,B_liste,k)
row1<-c(derive_lambda_carre(lambda,alpha,beta,A_liste,Liste_Hawkes,k),
        derive_lambda_alpha(lambda,alpha,beta,A_liste,Liste_Hawkes,k),
        derive_lambda_beta(lambda,alpha,beta,A_liste,B_liste,Liste_Hawkes,k))
row2<-c(derive_lambda_alpha(lambda,alpha,beta,A_liste,Liste_Hawkes,k),
        derive_alpha_carre(lambda,alpha,beta,A_liste,Liste_Hawkes,k),
        derive_alpha_beta(lambda,alpha,beta,A_liste,B_liste,Liste_Hawkes,k))
row3<-c(derive_lambda_beta(lambda,alpha,beta,A_liste,B_liste,Liste_Hawkes,k),
        derive_alpha_beta(lambda,alpha,beta,A_liste,B_liste,Liste_Hawkes,k),
        derive_beta_carre(lambda,alpha,beta,A_liste,B_liste,C_liste,Liste_Hawkes,k))
hessienne<-rbind(row1,row2,row3)
hessienne
}

```

```

#Description: La fonction Fisher_matrix_fun calcule la matrice de Fisher
#et la matrice de variance covariance.
#Entrée      : La liste de points Liste_Hawkes
#            : Les paramètres inférés par=c(lambda,alpha,beta)
#            : Le nombre de points k.
#Sortie      : La matrice de Fisher et la matrice de variance-covariance.
Fisher_matrix_fun<-function(Liste_Hawkes,par,k){
  lambda<-par[1]
  alpha<-par[2]
  beta<-par[3]
  Fisher_matrix<-hessienne_fun(Liste_Hawkes,lambda,alpha,beta)
  Var_covar_matrix<-solve(Fisher_matrix)
  return(list(Fisher_matrix=Fisher_matrix,Var_covar_matrix=Var_covar_matrix))
}

```

```

#####Intervalles de confiances#####
#####
#Description: La fonction génère des intervalles de confiances à 95 pourcent pour
#les paramètres inférés.
#Entrée      : Les paramètres :par
#            : La matrice de variance covariance :Var_cov_matrix
#            : Le temps final :T ol sera toujours pris au temps de la dernière arrivée.
#Sortie      : Un tableau de taille 2*3 affichant les intervalles de confiances.
Confidence_intervalle<-function(par, Var_cov_matrix,T)
{intervalle_lambda<-c(lambda-1.96*sqrt(Var_cov_matrix[1,1]/T),
  lambda+1.96*sqrt(Var_cov_matrix[1,1]/T))
intervalle_alpha<-c(alpha-1.96*sqrt(Var_cov_matrix[2,2]/T),
  alpha+1.96*sqrt(Var_cov_matrix[2,2]/T))
intervalle_beta<-c(beta-1.96*sqrt(Var_cov_matrix[3,3]/T),
  beta+1.96*sqrt(Var_cov_matrix[3,3]/T))
}

```



```
cbind(intervalle_lamda,intervalle_alpha,intervalle_beta)
}
```

```
Var_covar_matrix<-Fisher_matrix_fun(Liste_Hawkes,par,k)$Var_covar_matrix
Confidence_intervalle(par,Var_covar_matrix,T)
```

8.7 Validation du modèle Hawkes

Code R 8.11. *Validation du modèle avec calcul des résidus (voir 5.5).*

```
#Description: Calcul la valeur du compensateur d'un processus de Hawkes au temps t.
#Entree:      Le temps t
#             : Les paramètres lambda, alpha et beta
#             : La liste des points du processus Liste_Hawkes.
#Sortie       : La valeur du compensateur.
compensateur_fun<-function(Liste_Hawkes,lambda,alpha,beta,t){
  N<-length(Liste_Hawkes[Liste_Hawkes<t])#Le nombre d'instants d'arrivées avant t
  Nratio<-alpha/beta
  res<-lambda*t
  if(N!=0){
    for (i in 1:N){
      res<-res+Nratio*(1-exp(-beta*(t-Liste_Hawkes[i])))
    }
  }
  return(res)
}

#Description: Fonction qui retourne les résidus d'un processus de Hawkes.
#Entrée       : Une liste de points ordonné: Liste.
#             : Les paramètres lambda alpha et beta.
#Sortie       : La liste des temps d'inter-arrivées du processus transformés
#par le compensateur.
residus_hawkes_fun<-function(Liste,lambda,alpha,beta){
  k<-length(Liste)
  res<-c()
  for (i in 1:k){
    res<-c(res,compensateur_fun(Liste,lambda,alpha,beta,Liste[i]))
  }
  res<-c(res[1],diff(res))
  #diff calcule la différence terme à terme: t_i-t_{i-1} pour tout i>=2.
  return(res)
}

Liste<-simu_hawkes_inversion(0.5,1,1.5,100)
residus_hawkes<-residus_hawkes_fun(Liste,0.5,1,1.5)
ks.test(residus_hawkes,'pexp',1)
```

8.8 Application du processus de Hawkes à des données simulées

```
#####
#####Application à des données simulées#####
Liste_Hawkes<-simu_hawkes_inversion(0.5,1,1.5,150)
mle_Hawkes_fun(Liste_Hawkes,NULL) #Utiliser la fonction mle_hawkes
mle_Hawkes_simulate<-mle_Hawkes_fun(Liste_Hawkes,c(1,0.1,0.1))
Hessienne_opt<-mle_Hawkes_simulate$Matrice_hessienne #La matrice Hessienne de l'optimiseur optim
par<-mle_Hawkes_simulate$param
lambda<-par[1]
alpha<-par[2]
beta<-par[3]
plot(lambda_star,xlim=c(0,Liste_Hawkes[length(Liste_Hawkes)]),n=10**4)
#L'intensité conditionnelle ajustée
Hessienne_exacte<-hessienne_fun(Liste_Hawkes,lambda,alpha,beta)
Hessienne_exacte
#La matrice Hessienne théorique.
Hessienne_exacte-Hessienne_opt
#L'écart entre la matrice fournie par optim et la matrice exacte.
norm(Hessienne_exacte-Hessienne_opt,"1")
Fisher_matrix_fun(Liste_Hawkes,par,k)
#La matrice de Fisher et la matrice de variance covariance
eigen(Fisher_matrix_fun(Liste_Hawkes,par,k)$Fisher_matrix)$values
#Les valeurs propres de la matrice de Fisher.
eigen(Fisher_matrix_fun(Liste_Hawkes,par,k)$Fisher_matrix)$values>0
#Définie positive.
Var_covar_matrix<-Fisher_matrix_fun(Liste_Hawkes,par,k)$Var_covar_matrix
Confidence_intervalle(par,Var_covar_matrix,T)
#Les intervalles de confiance à 95 pourcent des paramètres
#####
residus2012<-residus_hawkes_fun(Liste_Hawkes,lambda,alpha,beta)
hist(residus2012)
ks.test(residus2012,'pexp',1)
#Etude des résidus
```

8.9 Application du processus de Hawkes aux données de libération de spores

```
Liste_Hawkes<-Temps2011#Appeler sa liste Liste_Hawkes
#Utiliser la fonction mle_hawkes
mle_Hawkes2011<-mle_Hawkes_fun(Liste_Hawkes,NULL)
mle_Hawkes2011
Hessienne_opt<-mle_Hawkes2011$Matrice_hessienne #La matrice Hessienne de l'optimiseur optim
par<-mle_Hawkes2011$param
lambda<-par[1]
```

```

alpha<-par[2]
beta<-par[3]
plot(lambda_star,xlim=c(0,Liste_Hawkes[length(Liste_Hawkes)]),n=10**4)
#L'intensité conditionnelle ajustée.
Hessienne_exacte<-hessienne_fun(Liste_Hawkes,lambda,alpha,beta)
Hessienne_exacte #La matrice Hessienne théorique.
Hessienne_exacte-Hessienne_opt
#L'écart entre la matrice fournie par optim et la matrice exacte.
norm(Hessienne_exacte-Hessienne_opt,"1")
Fisher_matrix_fun(Liste_Hawkes,par,k)
#La matrice de Fihser et la matrice de variance covariance
eigen(Fisher_matrix_fun(Liste_Hawkes,par,k)$Fisher_matrix)$values
#Les valeurs propres de la matrice de Fisher.
eigen(Fisher_matrix_fun(Liste_Hawkes,par,k)$Fisher_matrix)$values>0
#Définie positive.
Var_covar_matrix<-Fisher_matrix_fun(Liste_Hawkes,par,k)$Var_covar_matrix
Confidence_intervalle(par,Var_covar_matrix,T)
#Les intervalles de confiance à 95 pourcent des paramètres
residus2011<-residus_hawkes_fun(Liste_Hawkes,lambda,alpha,beta)
hist(residus2011)
ks.test(residus2011,'pexp',1)#test de kolmogorov smirnoff d'adéquation avec
#la loi exponentielle.
#Etude des residus

#Pour2012:
Liste_Hawkes<-Temps2012
#Utiliser la fonction mle_hawkes
mle_Hawkes2012<-mle_Hawkes_fun(Liste_Hawkes,NULL)
mle_Hawkes2012
Hessienne_opt<-mle_Hawkes2011$Matrice_hessienne #La matrice Hessienne de l'optimiseur optim
par<-mle_Hawkes2011$param
lambda<-par[1]
alpha<-par[2]
beta<-par[3]
plot(lambda_star,xlim=c(0,Liste_Hawkes[length(Liste_Hawkes)]),n=10**4)
#L'intensité conditionnelle ajustée
Hessienne_exacte<-hessienne_fun(Liste_Hawkes,lambda,alpha,beta)
Hessienne_exacte
#La matrice Hessienne théorique.
Hessienne_exacte-Hessienne_opt
#L'écart entre la matrice fournie par optim et la matrice exacte.
norm(Hessienne_exacte-Hessienne_opt,"1")
Fisher_matrix_fun(Liste_Hawkes,par,k)
#La matrice de Fihser et la matrice de variance covariance
eigen(Fisher_matrix_fun(Liste_Hawkes,par,k)$Fisher_matrix)$values
#Les valeurs propres de la matrice de Fisher.

```

```

eigen(Fisher_matrix_fun(Liste_Hawkes,par,k)$Fisher_matrix)$values>0
#Définie positive.
Var_covar_matrix<-Fisher_matrix_fun(Liste_Hawkes,par,k)$Var_covar_matrix
Confidence_intervalle(par,Var_covar_matrix,T)
#Les intervalles de confiance à 95 pourcent des paramètres
#####
residus2012<-residus_hawkes_fun(Liste_Hawkes,lambda,alpha,beta)
hist(residus2012)
ks.test(residus2012,'pexp',1)
#Etude des résidus

```

8.10 Préparation des données

```

#Les codes relatifs aux données météo sont présentés ici.
meteo<-read.csv("Meteo.csv",sep=";",header=TRUE)
#Les variables codées en type caractère 'chr' sont transformées
#en type numérique num:

library("reshape2")
meteo$PRECIPIT<-type.convert(meteo$PRECIPIT,dec=";",as.is=TRUE)
meteo$AIRTEMP<-type.convert(meteo$AIRTEMP,dec=";",as.is=TRUE)
meteo$SOILTEMP1<-type.convert(meteo$SOILTEMP1,dec=";",as.is=TRUE)
meteo$TEMP1<-type.convert(meteo$TEMP1,dec=";",as.is=TRUE)
meteo$WINDSPEED2<-type.convert(meteo$WINDSPEED2,dec=";",as.is=TRUE)
summary(meteo)

#On sépare les données en deux années distinctes
meteo2011<-subset(meteo,meteo$Annee=='2011')
meteo2012<-subset(meteo,meteo$Annee=='2012')

#On sélectionne seulement la plage de temps dans laquelle
#la libération de spores a été réalisée.

meteo2011<-meteo[11161:18240,]
meteo2012<-meteo[56161:63717,]

#On choisit de regrouper les cinq mesures par heures
#pour avoir une mesure par heure.
#Or la liste de points de 2012 possède 7557 mesures
#qui n'est pas multiple de cinq.
#On peut supposer qu'il existe des heures avec plus de
#cinq mesures et d'autres avec moins de cinq mesures.
#La fonction transfo regroupe les valeurs par heure, en tenant compte de cela.
#Le critère est de prendre le maximum des mesures sur l'heure.
Nombre_lignes2011<-length(meteo2011$Annee)#7080

```

```
Nombre_lignes2012<-length(meteo2012$Annee)#7557
```

```
#Description: transforme la liste de données avec environ cinq mesures par
#heures en une liste plus petite possédant une mesure par heure.
#Entrée: Une liste meteo_var correspondant à une covariable du tableau meteo2011
#Sortie: Une liste avec des mesures par heures.
transfo2011<-function(meteo_var){
  var_res<-c()#La liste à retourner
  val=c(meteo_var[1]) #Le liste des valeurs de cette variable sur une heure.
  for (i in 2:Nombre_lignes2011){
    if(meteo2011$Heure[i]!=meteo2011$Heure[i-1]){
      var_res<-c(var_res,max(val))#Si l'heure change on ajoute le maximum
      #des mesures de l'heure précédente dans var_res.
      val<-c()
    }
    val<-c(val,meteo_var[i])#liste des mesures sur une heure.
  }
  tail(meteo2011,n=5)#En 2011 les 5 dernières mesures sont bien dans la même heure.
  val<-max(tail(meteo_var,n=5))
  var_res<-c(var_res,val)
  return(var_res)
}

#Toutes les variables sont ainsi transformées.
PRECIPIT2011<-transfo2011(meteo2011$PRECIPIT)
AIRTEMP2011<-transfo2011(meteo2011$AIRTEMP)
RELHUMID2011<-transfo2011(meteo2011$RELHUMID)
BRIGHTNESS2011<-transfo2011(meteo2011$BRIGHTNESS)
LEAFWET12011<-transfo2011(meteo2011$LEAFWET1)
SOILTEMP12011<-transfo2011(meteo2011$SOILTEMP1)
TEMP12011<-transfo2011(meteo2011$TEMP1)
SOLAR2011<-transfo2011(meteo2011$SOLAR)
WINDDIR2011<-transfo2011(meteo2011$WINDDIR)
AIRPRES2011<-transfo2011(meteo2011$AIRPRES)
WINDSPEED2011<-transfo2011(meteo2011$WINDSPEED2)

nombre_heure2011<-length(PRECIPIT2011)
Liste_heure2011<-1:nombre_heure2011
#Un data frame est créé avec les nouvelles covariables.
DonneeMeteo2011<-data.frame(Heure=Liste_heure2011,AIRTEMP2011,RELHUMID2011
                           ,BRIGHTNESS2011,LEAFWET12011,PRECIPIT2011
                           ,SOILTEMP12011,TEMP12011,SOLAR2011,WINDDIR2011
                           ,AIRPRES2011,WINDSPEED2011)

##on exécute la même procédure pour l'année 2012:

transfo2012<-function(meteo_var){
```

```

var_res<-c()
val=c(meteo_var[1])
for (i in 2:Nombre_lignes2012){
  if(meteo2012$Heure[i]!=meteo2012$Heure[i-1]){
    var_res<-c(var_res,max(val))
    val<-c()
  }
  val<-c(val,meteo_var[i])
}
tail(meteo2011,n=5)#En 2012 les 5 dernières valeurs sont bien dans la même heure.
val<-max(tail(meteo_var,n=5))
var_res<-c(var_res,val)
}

AIRTEMP2012<-transfo2012(meteo2012$AIRTEMP)
RELHUMID2012<-transfo2012(meteo2012$RELHUMID)
BRIGHTNESS2012<-transfo2012(meteo2012$BRIGHTNESS)
LEAFWET12012<-transfo2012(meteo2012$LEAFWET1)
PRECIPIT2012<-transfo2012(meteo2012$PRECIPIT)
SOILTEMP12012<-transfo2012(meteo2012$SOILTEMP1)
TEMP12012<-transfo2012(meteo2012$TEMP1)
SOLAR2012<-transfo2012(meteo2012$SOLAR)
WINDDIR2012<-transfo2012(meteo2012$WINDDIR)
AIRPRES2012<-transfo2012(meteo2012$AIRPRES)
WINDSPEED2012<-transfo2012(meteo2012$WINDSPEED2)

nombre_heure2012<-length(AIRTEMP2012)
Liste_heure2012<-1:nombre_heure2012#Liste des heures cumulées
DonneeMeteo2012<-data.frame(Heure=Liste_heure2012,AIRTEMP2012,RELHUMID2012
                           ,BRIGHTNESS2012,LEAFWET12012,PRECIPIT2012
                           ,SOILTEMP12012,TEMP12012,SOLAR2012,WINDDIR2012
                           ,AIRPRES2012,WINDSPEED2012)

#On ajoute à ces tableaux les listes d'éjections de spores
#contenues dans le fichier askospory.csv
data<-read.csv("Askospory.csv",sep="," ,header=TRUE)#Le data frame
#contenant les instants et l'amplitude des libérations de spores.
Year2011<-subset(data,data$Annee=='2011')
Year2012<-subset(data,data$Annee=='2012')
Amplitude2011<-Year2011$N #L'amplitude pour chaque année
Amplitude2012<-Year2012$N

#Liberation2011 contient un indicateur de libération de spores.
#Temps2011 contient les instants de libérations de spores
Liberation2011<-rep(0,nombre_heure2011)
Temps2011<-c()

```

```

for (i in 1:nombre_heure2011){
  if (Amplitude2011[i]!=0){
    Liberation2011[i]<-1
    Temps2011<-c(Temps2011,i)
  }
}

Liberation2012<-rep(0,nombre_heure2012)
Temps2012<-c()
for (i in 1:nombre_heure2012){
  if (Amplitude2012[i]!=0){
    Liberation2012[i]<-1
    Temps2012<-c(Temps2012,i)
  }
}

DonneeMeteo2011$Amplitude<-Amplitude2011#On place les colonnes Liberation et Amplitude dans
#le tableau des données météo.
DonneeMeteo2012$Amplitude<-Amplitude2012

DonneeMeteo2011$Tempsliberation<-Liberation2011
DonneeMeteo2012$Tempsliberation<-Liberation2012

#On renomme les colonne de chaque tableau pour pouvoir les concaténer.
colnames(DonneeMeteo2011)<-c("Heure","AIRTEMP","RELHUMID","BRIGHTNESS"
                             ,"LEAFWET","PRECIPIT","SOILTEMP1","TEMP1","SOLAR"
                             ,"WINDDIR","AIRPRES","WINDSPEED","Amplitude"
                             ,"Liberation")
colnames(DonneeMeteo2012)<-c("Heure","AIRTEMP","RELHUMID","BRIGHTNESS"
                             ,"LEAFWET","PRECIPIT","SOILTEMP1","TEMP1","SOLAR"
                             ,"WINDDIR","AIRPRES","WINDSPEED","Amplitude"
                             ,"Liberation")

#Finalement on regroupe toutes ces varaibles dans un seul tableau commun appelé
#data_everything.
data_everything<-rbind(DonneeMeteo2011,DonneeMeteo2012)
#L'heure est modifiée pour qu'elle soit linéaire sur les deux années.
for (i in 1417:2928){#Réalise le cumul des heures sur les deux années.
  data_everything$Heure[i]<-data_everything$Heure[i]+1416
}
colAnnee<-c(rep("Annee2011",1416),rep("Annee2012",1512))
#Ajoute un indicateur de l'année
data_everything$Annee<-colAnnee
#Ajouter un effet année (1 en 2011, 0 sinon):
ListeA2011<-c(rep(1,1416),rep(0,1512))
data_everything$ListeA2011<-ListeA2011

```

```
#Instruction qui enregistre le tableau data_everything:
#save(data_everything,file="Data_everything.Rdata")
#Instruction qui enregistres les instants de libérations de spores:
#save(Temps2011,file="Temps2011.Rdata")
#save(Temps2012,file="Temps2011.Rdata")
```

8.11 Ajustement au processus de Poisson non homogène avec une intensité log-linéaire

#Etude des données météo avec le processus de Poisson non homogène avec une intensité log-linéaire.

```
#On désigne par T le nombre d'heures totales.
T<-2928
#La fonction glm de R estime les paramètres de la regression log-lineaire
#correspondant au maximum de vraisemblance pour le modèle de Poisson non
#homogène avec une intensite log-linéaire.
G<-glm(data_everything$Liberation~data_everything$LEAFWET
      +data_everything$RELHUMID
      +data_everything$BRIGHTNESS
      +data_everything$PRECIPIT
      +data_everything$TEMP1
      +data_everything$SOLAR
      +ListeA2011
      ,family=poisson(link = "log"))
summary(G)
logLik(G) #Maximum de vraisemblance trouvé par glm

#On peut vérifier qu'ils correspondent au maximum réel avec les fonctions
#suivantes.
#On crée quelques variables auxiliaires: Temps et Matrice.
Temps<-c()#La liste des instants de libération pour les deux années concaténées.
for (i in 1:T){
  if (data_everything$Liberation[i]!=0){
    Temps<-c(Temps,i)
  }
}
lambda_background<-rep(1,2928)#une intensité d'arrière plan constante. Une liste
#de 1.
Matrice<-cbind(lambda_background,#La matrice des covariables utilisées.
               data_everything$LEAFWET,
               data_everything$RELHUMID,
               data_everything$BRIGHTNESS,
               data_everything$PRECIPIT,
```



```

        data_everything$TEMP1,
        data_everything$SOLAR,
        ListeA2011)

#Description: fonction de vraisemblance pour le modèle log-lineaire.
#Entrée      : Les paramètres par.
#            : La liste des instants d'arrivées: Temps.
#            : La matrice des covariables avec un intercept
#            : une borne de temps T
mle_NHPlog<-function(par,Temps,Matrice,T)
{#x ext le vecteur des paramètre x=(beta_1,beta_2,...,beta_p) avec
  #p le nombre de paramètres.
  res<-0
  aux<-rep(0,T)
  for (t in 1:T){
    aux[t]<-sum(Matrice[t,]*t(par))
    res<-res+exp(aux[t])
  }
  res<- -res
  for (i in Temps){
    res<-res+aux[i]
  }
  return(res)
}
par<-G$coefficients
mle_NHPlog(par,Temps,Matrice,2928)#La vraisemblance est égale à -866:
logLik(G)

#Il est également possible d'écrire des fonctions pour le gradient ou la Hessienne.

#Description: calcul le gradient de l'estimateur du maximum de vraisemblance.
#Entrée: les paramètres par, la liste des instants d'arrivées Temps,
#la matrice des
#covariables avec un intercept Matrice et une borne de temps T.
grad_mle_NHPlog<-function(par,Temps,Matrice,T)
{ res<-rep(0,length(par))
  for (j in 1:length(par))
  {
    aux<-rep(0,T)
    for (t in 1:T){
      aux[t]<-sum(Matrice[t,]*t(par))
      res[j]<-res[j]+Matrice[t,j]*exp(aux[t])
    }
    res[j]<- -res[j]
    for (i in Temps){
      res[j]<-res[j]+Matrice[i,j]
    }
  }
}

```

```

    }
  }
  return(res)
}
grad_mle_NHPlog(par,Temps,Matrice,T) #le gradient est très faible.

#Description: La matrice hessienne de l'estimateur du maximum de vraisemblance.
#Entrée: les paramètres par, la matrice des covariables
#avec un intercept: Matrice.
#Une borne de temps T.
hessienne_mle_NHPlog<-function(par,Matrice,T)
{
  res<-matrix(rep(0,length(par)**2),length(par),length(par))
  for (j in 1:length(par))
  { for (k in 1:length(par))
    {aux<-rep(0,T)
     for (t in 1:T){
       aux[t]<-sum(Matrice[t,]*t(par))
       res[j,k]<-res[j,k]+Matrice[t,j]*Matrice[t,k]*exp(aux[t])
     }
     res[j,k]<- -res[j,k]
   }
  }
  return(res)
}
H<-hessienne_mle_NHPlog(par,Matrice,T)
FisherPPNH<--H
Matrice_variance_covariancePPNH<-solve(FisherPPNH)
sqrt(Matrice_variance_covariancePPNH[1])#l'écart type 'std. Error' de intercept.
summary(G)# La même valeur.

#Description: Calcule l'intensité sur chaque heure.
#Entée      : Les paramètres par et la matrice des covariables avec un intercept.
#           : La borne de temps T.
#Sortie     : Une liste de taille T contenant les intensité
#estimées pour chaque heure.
intensity_fun<-function(par,Matrice,T){
  aux<-rep(0,T)
  for (t in 1:T){
    aux[t]<-sum(Matrice[t,]*t(par))
  }
  intensite<-exp(aux)
  return(intensite)
}

```

```

Liste_intensite<-intensity_fun(par,Matrice,T)
par(mfrow=c(2,1))
plot.ts(Liste_intensite,xlim=c(0,300),ylim=c(0,1),col='blue',lwd=3)#l'intensite
#estimée.
plot(data_everything$Liberation,xlim=c(0,300),col='red')#les
#instants de libération de spores.
par(mfrow=c(1,1))
#La représentation ci-dessus correspond au nombre de libération par heure.
#Il est possible de regrouper les valeurs par jour. On compte le nombre de spores
#mesurée la journée et on le compare à la somme des intensité prédites
#le même jour.

#Lignes de codes pour compter le nombre de spores enregistrés par jour. La liste
#Liberation_day contient le nombre de spores réel libéré chaque jour.
Liberation_day<-rep(0,122)
j<-1
aux<-0
for(i in 1:2928){
  if (aux==24){
    aux<-0
    j<-j+1
  }
  Liberation_day[j]<-Liberation_day[j]+data_everything$Liberation[i]
  aux<-aux+1
}

#Lignes de codes pour calculer l'intensité estimée par jour. Elle est placée
#dans la liste appelée Intensity_day.

Intensity_day<-rep(0,122)
j<-1
aux<-0
for(i in 1:2928){
  if (aux==24){
    aux<-0
    j<-j+1
  }
  Intensity_day[j]<-Intensity_day[j]+Liste_intensite[i]
  aux<-aux+1
}

#Les 59 premiers jours correspondent à l'année 2011. On trace les deux graphiques:
#Nombre de spores réels et nombre de spores estimés. #la courbe rouge est la
#courbe des données, la bleue la courbe ajustée.
Liberation_day2011<-Liberation_day[1:59]
Intensity_day2011<-Intensity_day[1:59]
plot(Liberation_day2011,col='red',type='s',xlim=c(0,59),ylim=c(0,20),xlab="")

```

```

      ,ylab="")
par(new=TRUE)
plot(Intensity_day2011,col='blue',type='s',xlim=c(0,59),ylim=c(0,20),xlab=
      "Temps en jours (2011)", ylab="",lty=2)

#Le même graphique pour 2012
Liberation_day2011<-Liberation_day[60:122]
Intensity_day2011<-Intensity_day[60:122]
plot(Liberation_day2011,col='red',type='s',xlim=c(0,63),ylim=c(0,20),xlab=""
      ,ylab="")
par(new=TRUE)
plot(Intensity_day2011,col='blue',type='s',xlim=c(0,63),ylim=c(0,20),xlab=
      "Temps en jours (2012)", ylab="",lty=2)

#Description: Simulation du processus de Poisson avec une intensité log-linéaire
#par amincissement.
#Entrée: Intensité :Liste de taille T entier où intensite[i] est l'intensité sur
#l'heure i.
#Sortie: Une liste de points ordonnés sur (0,T)
simu_PPNH_log_thinning<-function(intensite,T){
  M<-max(intensite)
  Liste_aux<-simu_PPH(M,T)
  N<-length(Liste_aux)
  U<-runif(N,0,M)
  Liste<-c()
  for (i in 1:N){
    if(U[i]<=intensite[floor(Liste_aux[i])+1]){
      Liste<-c(Liste,Liste_aux[i])#Le premier element de la liste intensite est
      #dans la case numero 1.
    }
  }
  return(Liste)
}

simulation_PPNH_log<-simu_PPNH_log_thinning(Liste_intensite,2928)
#Graphique de l'intensité estimée en bleu avec les réels instants de
#libérations en rouge et les instants de libérations simulés par amincissement
#en marron.
plot(simulation_PPNH_log,1:length(simulation_PPNH_log)*c(0),xlim=c(0,300)
      ,col='brown',ylim=c(-0.1,0.3),xaxt="n",yaxt="n",xlab="",ylab="")
par(new=TRUE)
plot.ts(Liste_intensite,xlim=c(0,300),ylab="Intensite estimée",col='blue',lwd=2)
par(new=TRUE)
plot(Temps2011,1:length(Temps2011)*c(0),ylim=c(0,1),col='red'
      ,xlim=c(0,300),xaxt="n",yaxt="n",xlab="",ylab="")

```

