



Opéré au sein de :
l'Institut Elie Cartan

Ecole des Mines de Nancy

Loubna Ben Allal

Processus de Hawkes

Modélisation, Simulation et Inférence

Travail encadré par :
Antoine Lejay
Ernesto Mordecki
Radu Stoica

23 septembre 2020

Table des matières

1	Introduction	3
I	Parcours recherche : Processus de Hawkes	4
2	Processus de Poisson homogène	4
3	Processus de Poisson non homogène	8
4	Simulation des processus de Poisson	10
4.1	Processus de Poisson homogène	10
4.2	Processus de Poisson non homogène	11
4.2.1	Simulation par inversion	11
4.2.2	Simulation par amincissement	12
5	Inférence statistique sur les processus de Poisson homogènes	13
6	Processus de Hawkes	16
7	Simulation des processus de Hawkes	19
7.1	Algorithme d'amincissement classique	19
7.2	Algorithme d'amincissement modifié d'Ogata	21
7.3	Test des algorithmes de simulation des processus de Hawkes	22
7.3.1	Application au cas du processus de Poisson homogène	22
7.3.2	Comparaison des deux algorithmes	24
8	Inférence statistique sur les processus de Hawkes	26
8.1	La fonction de vraisemblance	26
8.2	Le gradient et la matrice hessienne	30
8.3	Estimation par maximisation de la vraisemblance	32
9	Validation du modèle	34
10	Applications	41
10.1	Application à des données simulées	41
10.2	Application à des données sismiques réelles	45
10.2.1	Présentation des données	45
10.2.2	Estimation des paramètres	46
10.2.3	Validation du modèle	49
II	Stage à l'IECL : Application à des modèles sismiques	51

11	Modèle de la fonction puissance	51
11.1	Modélisation	51
11.2	Simulation	52
11.3	Inférence statistique	53
11.4	Validation du modèle	59
11.5	Application à des données simulées	62
11.6	Application à des données réelles	64
12	Processus de Hawkes marqués	64
12.1	Modélisation	64
12.2	Simulation	66
12.3	Inférence	67
12.3.1	Marques indépendantes du temps	67
12.3.2	Marques qui dépendent du temps	68
12.4	Modèle ETAS (Epidemic Type Afterschock Sequence model) . .	70
12.4.1	Définition et modélisation	70
12.4.2	Simulation	71
12.4.3	Inférence	74
12.4.4	Package SAPP	76
12.4.5	Validation du modèle	86
12.4.6	Application à des données simulées	86
12.4.7	Application à des données réelles	86
13	Comparaison des modèles	89
13.1	Modèle exponentiel	89
13.2	Modèle puissance	91
13.3	Modèle ETAS	95
13.4	Conclusion	95
14	Processus de Hawkes spatio-temporels	96
14.1	Modélisation	96
14.2	Simulation	98
14.3	Inférence	102
15	Processus de Hawkes inhomogènes	103
15.1	Modèle ETAS non stationnaire	103
16	Conclusion	104

1 Introduction

Les processus de Hawkes sont des processus stochastiques qui ont des applications dans de différents domaines comme la sismologie, la finance et les réseaux sociaux. Ce sont des processus auto-excités, ce qui signifie que chaque événement arrivé augmente le taux des arrivées futures pour une certaine période de temps. C'est le cas pour les répliques des tremblements de terre ; un séisme augmente la tension géographique dans la région, ce qui peut causer d'autres secousses. C'est aussi le cas pour certains événements en finance, tel que la faillite d'une grande banque d'investissement qui peut envoyer des chocs dans les centres du monde financier. Dans ce travail, on s'intéresse à la modélisation des processus de Hawkes, à leur simulation et à l'inférence statistique pour estimer les paramètres qui caractérisent ces processus.

Ce document est divisé en deux parties, la première concerne le travail qu'on a effectué pendant mon parcours recherche de l'Ecole des Mines de Nancy et pendant lequel on a traité les processus de Hawkes avec un modèle exponentiel. La deuxième partie concerne mon stage de 3 mois à l'Institut Elie Cartan, et pendant lequel on est passé au processus de Hawkes avec le modèle puissance et puis aux processus marqués notamment le modèle ETAS ainsi qu'une partie du modèle spatio-temporel.

On s'intéresse d'abord, dans les quatre premières sections, aux processus de Poisson, qui sont les processus de base où les événements arrivent indépendamment. On distingue deux types, les processus homogènes où le taux d'arrivée des événements est constant dans le temps, et les processus non homogènes pour lesquels ce taux varie. Ensuite, dans les sections 6 et 7, on présente la modélisation mathématique des processus de Hawkes, ainsi que des méthodes pour simuler ce type de processus. Dans la section 8, on s'intéresse à l'estimation des paramètres des modèles de Hawkes par maximisation de la vraisemblance. La section 9 présente des procédures pour valider les modèles. Dans les sections qui suivent, on a suivi la même approche pour le modèle puissance, et le modèle marqué.

Première partie

Parcours recherche : Processus de Hawkes

Cette première partie est constituée du travail que j'ai effectué pendant mon parcours recherche à Mines Nancy. Elle contient une présentation des processus de Hawkes en général et d'une étude des processus de Hawkes avec un modèle exponentiel.

2 Processus de Poisson homogène

On s'intéresse dans cette partie aux processus de Poisson homogènes

Définition 1. Soit $(\tau_i)_{i \geq 1}$ une suite de variables aléatoires i.i.d de loi exponentielle de paramètre $\lambda > 0$ et $T_n = \sum_{i=1}^n (\tau_i)$. Le processus $(N_t, t \geq 0)$ tel que $N_t = \sum_{i \geq 1} (1_{t \geq T_i})$ s'appelle un *processus de Poisson* d'intensité λ .

Ainsi, on peut interpréter $N(t)$ comme le nombre d'événements produits jusqu'à l'instant t sachant que ces événements surviennent à des dates séparées par des durées qui suivent une loi exponentielle de paramètre λ .

Proposition 1. Soit $(N_t, t \geq 0)$ un processus de Poisson défini comme ci-dessus, alors

$$\begin{aligned}\forall t, h > 0, \quad \mathbb{P}(N(t+h) - N(t) = 0) &= 1 - \lambda h + o(h), \\ \mathbb{P}(N(t+h) - N(t) = 1) &= \lambda h + o(h), \\ \mathbb{P}(N(t+h) - N(t) = n) &= o(h), \quad \forall n > 1.\end{aligned}$$

Lemme 1. Notons par τ_t la durée qu'il faut pour qu'un événement se produise à partir de la date t . Si on note t_k le dernier événement produit avant cette date, alors :

- i τ_t suit aussi une loi exponentielle de paramètre λ ,
- ii τ_t est indépendante des $(\tau_i)_{i \neq k}$.

Démonstration du lemme 1. Soit $h > 0$,

- i On a $\tau_k = \tau_t + t - t_k$

Comme τ_k suit une loi exponentielle alors c'est une variable sans mémoire, donc

$$\mathbb{P}(\tau_k > h + t - t_k | \tau_k > t - t_k) = \mathbb{P}(\tau_k > h)$$

D'où

$$\mathbb{P}(\tau_k > h) = \mathbb{P}(\tau_k - (t - t_k) > h | \tau_k - (t - t_k) > 0) = \mathbb{P}(\tau_t > h | \tau_t > 0) = \mathbb{P}(\tau_t > h).$$

Par suite τ_k et τ_t sont de même loi, ainsi τ_t suit une loi exponentielle de paramètre λ .

ii Soient $j \neq k$ et $n, m \in \mathbb{N}$. On a $\tau_k = \tau_t + t - t_k$ donc

$$\mathbb{P}(\tau_t = n, \tau_j = m) = \mathbb{P}(\tau_k = n - (t - t_k), \tau_j = m).$$

Comme τ_k et τ_j sont indépendantes alors

$$\mathbb{P}(\tau_t = n, \tau_j = m) = \mathbb{P}(\tau_k = n - (t - t_k))\mathbb{P}(\tau_j = m) = \mathbb{P}(\tau_t = n)\mathbb{P}(\tau_j = m).$$

D'où le résultat. □

Démonstration de la proposition 1. Pour h très petit, et d'après le lemme 1

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

Pour le deuxième cas, on le démontre ici pour $n = 2$, la preuve est similaire pour $n > 2$. Pour que 2 événements aient lieu entre t et $t + h$, il faut que la somme de τ_t et la durée d'arrivée du 2^{ème} événement soit inférieure à h

$$\mathbb{P}(N(t+h) - N(t) = 2) = \mathbb{P}((\tau_t + \tau_{k+1} \leq h).$$

Comme τ_{k+1} et τ_t sont deux variables aléatoires exponentielles de paramètre λ indépendantes, leur somme suit une loi Gamma de fonction de répartition :

$$F(s) = 1 - \exp(-\lambda s) - \lambda s \exp(-\lambda s), \quad \forall s > 0.$$

Donc

$$\begin{aligned} \mathbb{P}(N(t+h) - N(t) = 2) &= 1 - \exp(-\lambda h) - \lambda h \exp(-\lambda h) \\ &= 1 - (1 - \lambda h + o(h)) - \lambda h(1 - \lambda h + o(h)) = \lambda^2 h^2 = o(h). \end{aligned}$$

Traisons enfin le dernier cas :

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

Cela conclut la démonstration. □

Proposition 2. *i) Le processus de Poisson est à accroissements stationnaires, c-à-d $\forall t, h > 0$, $N(t+h) - N(t)$ ne dépend que de h .*

ii) Le processus de Poisson est à accroissements indépendants, c-à-d : $\forall t_0 < t_1 < \dots < t_n$ $N(t_1) - N(t_0), \dots, N(t_n) - N(t_{n-1})$ sont mutuellement indépendants.

Démonstration. i) découle de la proposition 1.

ii) Pour alléger les notations, on démontre l'indépendance entre deux accroissements, le résultat se démontre de la même façon pour l'indépendance mutuelle.

Soient m_1, m_2, j, k dans \mathbb{N} tq $j \neq k$:

$$\mathbb{P}(N(t_j) - N(t_{j-1}) = m_1 \cap N(t_k) - N(t_{k-1}) = m_2) = \mathbb{P}(A_1 \cap A_2)$$

avec

$$A_1 = \left(\sum_{i=1}^{m_1} (\tau_{t_{j+i-1}} \leq (t_j - t_{j-1})) \right) \cap \left(\sum_{i=1}^{m_1+1} (\tau_{t_{j+i-1}} > (t_j - t_{j-1})) \right),$$

$$A_2 = \left(\sum_{i=1}^{m_2} (\tau_{t_{k+i-1}} \leq (t_k - t_{k-1})) \right) \cap \left(\sum_{i=1}^{m_2+1} (\tau_{t_{k+i-1}} > (t_k - t_{k-1})) \right).$$

Par indépendance des τ_i , A_1 et A_2 sont indépendants, donc

$$\begin{aligned} \mathbb{P}(N(t_j) - N(t_{j-1}) = m_1 \cap N(t_k) - N(t_{k-1}) = m_2) &= \mathbb{P}(A_1) \mathbb{P}(A_2) \\ &= \mathbb{P}(N(t_j) - N(t_{j-1}) = m_1) \mathbb{P}(N(t_k) - N(t_{k-1}) = m_2). \end{aligned}$$

D'où le résultat. \square

Théorème 1. Soit $(N_t, t \geq 0)$ un processus de Poisson de paramètre λ , alors le nombre d'événements produits dans un intervalle de durée t est une variable aléatoire qui suit une loi de poisson de paramètre λt :

$$P(N(s+t) - N(s) = n) = \exp(-\lambda t) \frac{(\lambda t)^n}{n!}, \quad \forall n \in \mathbb{N}, \quad \forall s > 0.$$

Lemme 2. On note $\mathbb{P}(N(t) = n)$ par $P_n(t)$. On a

$$\frac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t), \quad \forall n \geq 1.$$

Démonstration du lemme 2. Soit $h > 0$ très petit et $n \geq 1$. Par la formule des probabilités totales

$$\begin{aligned} P_n(t+h) &= \sum_{k=0}^n (P_{n-k}(t) \mathbb{P}(N(t+h) = n | N(t) = n-k)) \\ &= \sum_{k=0}^n (P_{n-k}(t) \mathbb{P}(N(t+h) - N(t) = k | N(t) = n-k)). \end{aligned}$$

Les accroissements du processus de Poisson sont stationnaires ainsi $N(t+h) - N(t)$ ne dépend que de h , donc cette quantité est indépendante de la valeur de $N(t)$, d'où :

$$\begin{aligned}
P_n(t+h) &= \sum_{k=0}^n (P_{n-k}(t) \mathbb{P}(N(t+h) - N(t) = k)) \\
&= P_n(t) \mathbb{P}(N(t+h) - N(t) = 0) + P_{n-1}(t) \mathbb{P}(N(t+h) - N(t) = 1) \\
&\quad + \sum_{k=2}^n (P_{n-k}(t) \mathbb{P}(N(t+h) - N(t) = k)) \\
&= P_n(t)(1 - \lambda h + o(h)) + P_{n-1}(t)(\lambda h + o(h)) + o(h) \\
&= P_n(t) - \lambda h P_n(t) + \lambda h P_{n-1}(t) + o(h).
\end{aligned}$$

Cette dernière ligne résulte de la proposition 1.

Donc

$$\frac{P_n(t+h) - P_n(t)}{h} = -\lambda P_n(t) + \lambda P_{n-1}(t) + \frac{o(h)}{h}.$$

Par passage à la limite $\frac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t)$.

Ceci conclut la démonstration du lemme. \square

Démonstration du théorème 1. On a

$$\frac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t), \quad \forall n \geq 1.$$

Pour le cas $n = 0$,

$$P_0(t+h) = P_0(t) \mathbb{P}(N(t+h) - N(t) = 0) = P_0(t)(1 - \lambda h + o(h)).$$

Donc on trouve

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t).$$

Ainsi nous disposons du système suivant :

$$\frac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t), \quad \forall n \geq 1, \quad (1)$$

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t), \quad (2)$$

$$P_0(0) = 1 \quad (3)$$

car $N(0) = 0$. Montrons par récurrence que

$$P_n(t) = \exp(-\lambda t) \frac{(\lambda t)^n}{n!}. \quad (4)$$

Pour $n = 0$, d'après (2), $P_0(t) = C \exp(-\lambda t)$ et d'après (3) $C = 1$ donc $P_0(t) = \exp(-\lambda t)$.

Supposons (4) vraie au rang n et montrons qu'elle est vraie au rang $n + 1$.
D'après (1),

$$\frac{dP_{n+1}(t)}{dt} + \lambda P_{n+1}(t) = \lambda P_n(t) = \lambda \exp(-\lambda t) \frac{(\lambda t)^n}{n!}. \quad (5)$$

Pour résoudre cette équation différentielle nous allons utiliser la méthode de variation de la constante :

$$P_{n+1}(t) = f(t) \exp(-\lambda t)$$

implique

$$f'(t) = \frac{\lambda(\lambda t)^n}{n!}.$$

Donc

$$f(t) = \frac{\lambda(\lambda t)^{n+1}}{(n+1)!} + A.$$

Ainsi

$$P_{n+1}(t) = \left(\frac{\lambda(\lambda t)^{n+1}}{(n+1)!} + A \right) \exp(-\lambda t).$$

Comme $P_{n+1}(0) = 0$ alors $A = 0$.

Et par suite

$$P_{n+1}(t) = \exp(-\lambda t) \frac{(\lambda t)^{n+1}}{(n+1)!}.$$

D'où le résultat. □

3 Processus de Poisson non homogène

Définition 2. Un processus aléatoire $(N(t), t > 0)$ est un processus de comptage si :

- $N(0) = 0$,
- N est une fonction croissante, continue à droite, et fixe entre deux sauts,
- $\Delta N_t = N_t - N_{t-} \in \{0, 1\}$ pour tout $t \in \mathbb{R}^+$.

Définition 3. Soit $(N(t), t > 0)$ un processus de comptage tel que :

- Les accroissements du processus sont indépendants
- Pour tout $t, h > 0$,

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

$$\mathbb{P}(N(t+h) - N(t) = n) = o(h), \quad \forall n \geq 1,$$

$$\mathbb{P}(N(t+h) - N(t) = 1) = \lambda(t)h + o(h).$$

Alors ce processus est un processus de Poisson non homogène de fonction d'intensité $\lambda(t)$.

Proposition 3 (Passage du processus de Poisson homogène au non homogène). Soit $(M(t), t > 0)$ un processus de Poisson homogène de paramètre 1, et soit m une fonction de \mathbb{R} dans \mathbb{R} strictement croissante et dérivable qui s'annule en 0. Posons $N(t) = M(m(t))$, alors $(N(t), t > 0)$ est un processus de Poisson non homogène de fonction d'intensité $\lambda(t) = m'(t)$.

Démonstration. Montrons d'abord que les accroissements de M sont indépendants, soient $0 < t_0 < t_1 < t_2 < t_3$, on a

$$\begin{aligned} N(t_1) - N(t_0) &= M(m(t_1)) - M(m(t_0)) \\ N(t_3) - N(t_2) &= M(m(t_3)) - M(m(t_2)). \end{aligned}$$

Comme m est strictement croissante, donc $0 < m(t_0) < m(t_1) < m(t_2) < m(t_3)$. Par suite ces deux quantités sont indépendantes d'après l'indépendance des accroissements du processus homogène N .

Soit $h > 0$, comme $m(t+h) - m(t) = m'(t)h + o(h)$ d'après la proposition 1,

$$\begin{aligned} \mathbb{P}(N(t+h) - N(t) = 1) &= \mathbb{P}(M(m(t+h)) - M(m(t)) = 1) \\ &= m(t+h) - m(t) + o(h) = m'(t)h + o(h), \end{aligned}$$

$$\begin{aligned} \mathbb{P}(N(t+h) - N(t) = 0) &= \mathbb{P}(M(m(t+h)) - M(m(t)) = 0) \\ &= 1 - (m(t+h) - m(t)) + o(h) = 1 - m'(t)h + o(h). \end{aligned}$$

Pour $k > 1$

$$\mathbb{P}(N(t+h) - N(t) = k) = \mathbb{P}(M(m(t+h)) - M(m(t)) = k) = o(h)$$

Donc N est bien un processus de Poisson non homogène. \square

Remarque 1. On peut passer de même du processus de Poisson non homogène au processus homogène. En effet, si $(N(t), t > 0)$ est un processus de Poisson non homogène de fonction d'intensité λ et de fonction moyenne m définie par $m(t) = \int_0^t \lambda(s)ds$, m est continue et strictement croissante donc elle est bijective, soit m^{-1} sa fonction inverse.

Posons $M(t) = N(m^{-1}(t))$, alors $(M(t), t > 0)$ est un processus de Poisson homogène de paramètre 1.

Corollaire 1. Soit $(N(t), t > 0)$ un processus de Poisson non homogène de fonction d'intensité λ et de fonction moyenne m tel que $m(t) = \int_0^t \lambda(s)ds$, alors $N(t)$ suit une loi de Poisson de paramètre $m(t)$

Preuve du corollaire 1. D'après la proposition 3, on peut générer un processus N' non homogène de fonction moyenne m à partir d'un processus de Poisson homogène M de paramètre 1. Pour $t > 0$ $N'(t) = M(m(t))$, soit $n \in \mathbb{N}$, d'après le théorème 1, $M(s)$ suit une loi de Poisson de paramètre $s \forall s > 0$. On a :

$$\mathbb{P}(N(t) = n) = \mathbb{P}(M(m(t)) = n) = \exp(-m(t)) \frac{(m(t))^n}{n!}.$$

Les processus N et N' sont des processus de Poisson non homogènes de même fonction moyenne, donc ils ont la même intensité. Par suite $N(t)$ a la même loi que $N'(t)$ qui est une loi de Poisson de paramètre $m(t)$. Ce résultat découle du théorème 2 qui sera cité dans la section suivante.

Ce qui achève la démonstration. \square

4 Simulation des processus de Poisson

4.1 Processus de Poisson homogène

Pour simuler un processus de Poisson homogène de paramètre λ , on génère un vecteur de variables indépendantes, exponentiellement de paramètre λ . Il s'agit du vecteur des inter-arrivées $(\tau_k)_{k \geq 1}$, puis on génère le vecteur des occurrences $(t_k)_{k \geq 1}$ par la formule :

$$\begin{aligned} t_1 &= \tau_1, \\ t_k &= t_{k-1} + \tau_k, \quad \forall k > 1. \end{aligned}$$

On peut ensuite tracer la trajectoire de N en utilisant une fonction étagée, le code suivant donne le programme sur R

```
##Simulation en fonction du nombree d'evenements##
simpoisson=function(n,a) {
  T=cumsum(rexp(n,a))
  t=c(0,T)
  N=c(0:n)
  plot(t,N,type="s",ylab="N(t)")
  return(t)
}
#commande pour obtenir un processus de Poisson homogene de parametre 2
#et de taille 15:
simpoisson(15,2)

##Simulation jusqu'a une date T##
Poisson_homogene=function(T,a){
  i=2;t=0
  while (t[length(t)]<T){
    w=exp(1,a)
    t[i]=t[i-1]+w
    i=i+1
  }
  return(t) }
#commandes d'excution
a=2;T=20
t=Poisson_homogene(T,a)
#Trajectoire du processus
```

```

n=length(t)-1
N=c(0:n)
plot(t,N,type="s",ylab="N(t)")

```

4.2 Processus de Poisson non homogène

On simule les processus de Poisson non homogènes de deux façons, la première en utilisant la méthode de l'inverse de la fonction moyenne sur un processus de Poisson homogène, et la deuxième en utilisant un algorithme dit d'amin-cissement.

4.2.1 Simulation par inversion

Pour simuler un processus de Poisson non homogène de moyenne $m(t)$, on utilise la transformation de la proposition 3. On génère un vecteur des dates d'occurrences t_k d'un processus de Poisson homogène de paramètre 1. Et on obtient celui du processus non homogène par la formule $T_k = m^{-1}(t_k)$, le code suivant donne le programme de simulation sur R.

```

##Simulation en fonction du nombre d'evenements##
simpoisson2=function(n,minv) {
  T=simpoisson(n,1)
  t=minv(T)
  N=c(0:n)
  plot(t,N,type="s",ylab="N(t)")
}

#Commandes pour simuler un processus de Poisson non homogene
#de fonction moyenne m(t)=t^2
minv=function(t){sqrt(t)}
simpoisson2(15,minv)

##Simulation jusqu'un point T##
#processus de poisson non homogene de fonction d'intensite 2t
# et donc de fonction moyenne m(t)=t^2
Poisson_non_homogene=function(T,m_inverse) {
  t_hom=Poisson_homogene(T,1)
  t=m_inverse(t_hom)
  return(t)
}

#Commandes pour simuler un processus de poisson non homogene
#de fonction moyenne m(t)=t^2
m_inverse=function(t){sqrt(t)}
t=Poisson_non_homogene(100,m_inverse)

```

```

n=length(t)-1
N=c(0:n)
plot(t,N,type="s",ylab="N(t)",col="blue",main="Realization of a
homogeneous Poisson Process")

```

4.2.2 Simulation par amincissement

Théorème 2. *Considérons un processus de Poisson homogène $\bar{N}(t)$ d'intensité $\bar{\lambda}$. Soient $t_1, t_2, \dots, t_{N(T)}$ les dates d'occurrence du processus sur $(0, T]$. Supposons que $\forall t \in (0, T]$ $0 \leq \lambda(t) \leq \bar{\lambda}$. Pour $k = 1, \dots, N(T)$, on supprime le point t_k avec la probabilité $1 - \frac{\lambda(t_k)}{\bar{\lambda}}$; alors les points qui restent forment un processus de Poisson non homogène d'intensité $\lambda(t)$.*

Démonstration. Voir le théorème 4.2 à la page 8 de l'article de Yuanda Chen [2]. \square

D'après le théorème ci-dessus, pour simuler un processus de Poisson non homogène d'intensité $\lambda(t) \leq \bar{\lambda}, \forall t \geq 0$ avec $\bar{\lambda}$ un réel, sur un intervalle $(0, T]$, on peut suivre les étapes suivantes :

- On génère un processus de Poisson homogène $N(\cdot)$ d'intensité $\bar{\lambda}$. On note ces temps d'occurrence $t_1, t_2, \dots, t_{N(T)}$.
- Pour $k = 1, 2, \dots, N(T)$, on rejette t_k avec la probabilité $1 - \frac{\lambda(t_k)}{\bar{\lambda}}$.
- Les points restant forment un processus de Poisson non homogène d'intensité $\lambda(t)$.

Le programme ci-dessous donne le code de programmation sur R.

```

Poisson_non_homogene2=function(intensity_NH,lambda,T){
  #intensity_NH fonction d'intensite du processus
  t=s=0
  while (s[length(s)]<T){
    i=length(s)
    w=rexp(1,lambda)    #l'inter-arrivee
    s1=s[i]+w
    s=c(s,s1)           #actualiser le vecteur des dates du processus
                        #homogene
    u=runif(1,0,1)
    if (u<(intensity_NH(s1)/lambda)){ #on accepte s1 avec la probabilit
      intensity_NH(s1)/lambda
      t=c(t,s1)         #actualiser le vecteur des dates d'occurrences
    }
  }
  if (t[length(t)]<=T) {
    N=c(0:length(t))
    t=c(0,t)
  }
}

```

```

}
else{
  p=length(t)-1
  N=c(0:p)
  t=c(0,t[1:p])
}
list (time=t,nbpoints=N)
}
#commandes d'exécution pour un exemple
intensity_NH=function(t){t^2};T=4;lambda=16

set.seed(2020)

t_NH=Poisson_non_homogene2(intensity_NH,lambda,T)
plot(t_NH$time,t_NH$nbpoints,xlab="time",type="s",ylab="N(t)",col="blue",
     main="Realization of non homogeneous Poisson process")

```

Les figures au dessous donnent les trajectoires de deux processus de poisson. On remarque que pour le processus non homogène les événements se succèdent plus vite quand le temps augmente.

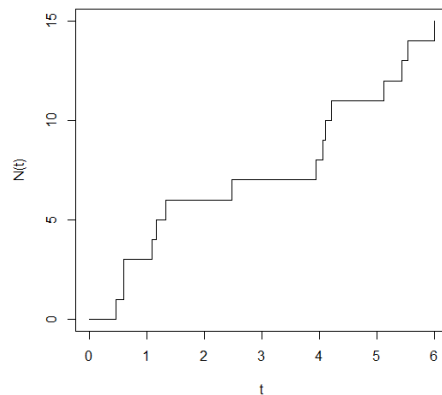


FIGURE 1 – Processus de Poisson homogène de paramètre $\lambda = 2$

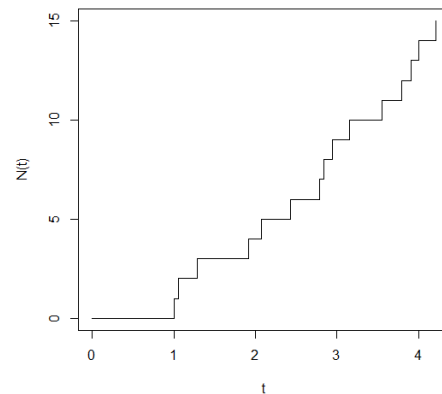


FIGURE 2 – Processus de Poisson non homogène de fonction moyenne $m(t) = t^2$

5 Inférence statistique sur les processus de Poisson homogènes

Dans cette section, on s'intéresse à l'estimation de l'intensité d'un processus de Poisson par la méthode de maximisation de la vraisemblance.

Soit $N(\cdot)$ un processus de poisson homogène d'intensité λ sur l'intervalle $[0, T]$, et soient t_1, t_2, \dots, t_k des dates d'arrivée des points du processus sur $[0, T]$.

Théorème 3 (Fonction de vraisemblance). *La fonction de vraisemblance de $N(\cdot)$, $L(t_1, t_2, \dots, t_k; \lambda)$ qu'on note L est :*

$$L = \lambda^k \exp(-\lambda T).$$

Démonstration. La fonction de vraisemblance est définie par :

$$\begin{aligned} L(t_1, t_2, \dots, t_k; \lambda) &= \mathbb{P}(T_1 = t_1, T_2 = t_2, \dots, T_k = t_k, T_{k+1} > T | \lambda) \\ &= f_{T_1, T_2, \dots, T_k}(t_1, t_2, \dots, t_k, T_{k+1} > T). \end{aligned}$$

Pour $i = 1, 2, \dots, k+1$, soient $W_i = T_i$ les variables des interarrivées, leurs réalisations sont $w_i = t_i - t_{i-1}$ avec la convention $T_0 = t_0 = 0$.

En appliquant la formule de changement de variables multidimensionnel, on obtient :

$$\begin{aligned} L &= f_{T_1, T_2, \dots, T_k}(t_1, t_2, \dots, t_k) \mathbb{P}(T_{k+1} > T) \\ &= f_{W_1, W_2, \dots, W_k}(w_1, w_2, \dots, w_k) |J|^{-1} \mathbb{P}(W_{k+1} > T - t_k). \end{aligned}$$

Avec J la matrice jacobienne telle que

$$J_{ij} = \frac{dT_i}{dW_j}.$$

Comme $T_k = W_1 + W_2 + \dots + W_k$, alors J est une matrice triangulaire inférieure dont les termes de la diagonale sont égaux à 1 et par suite $|J| = 1$. Donc

$$\begin{aligned} L &= f_{W_1, W_2, \dots, W_k}(w_1, w_2, \dots, w_k) \mathbb{P}(W_{k+1} > T - t_k) \\ &= \prod_{i=1}^k f_W(t_i - t_{i-1}) \exp(-\lambda(T - t_k)) \\ &= \lambda^k \exp(-\lambda \sum_{i=1}^k (t_i - t_{i-1})) \exp(-\lambda(T - t_k)) \\ &= \lambda^k \exp(-\lambda T). \end{aligned}$$

D'où le résultat. □

Proposition 4. *Un estimateur de maximum de vraisemblance de l'intensité λ du processus $N(\cdot)$ est*

$$\hat{\lambda} = \frac{k}{T}.$$

Démonstration. La log-vraisemblance de $N(\cdot)$ s'écrit

$$\ell = \log(L) = \log(\lambda^k \exp(-\lambda T)) = k \log(\lambda) - \lambda T.$$

```

> simpoisson(40,2)
[1] 0.0000000 0.2289677 0.7519711 0.9080111 1.3018012 2.2494373 2.7945527
[8] 4.0703047 4.4169990 4.9647233 5.0494635 5.3039407 6.5838190 7.0638244
[15] 7.3378187 8.1657407 8.2522365 10.5370751 15.4393296 15.9069641 16.2009687
[22] 16.4050783 17.5059937 17.5100702 17.5609566 18.1303339 18.2758227 18.6152294
[29] 18.9146882 19.3258714 19.4608158 19.8323437 20.0538728 20.0667882 20.2479314
[36] 20.4700403 21.0313981 21.0810993 21.9977885 22.0885764 22.5676333
> 40/22.5676333 #lambda estimée
[1] 1.77245

```

FIGURE 3 – estimation de l'intensité d'un processus de Poisson homogène

Le maximum est trouvé en résolvant l'équation :

$$\frac{\partial \ell}{\partial \lambda} = \frac{k}{\lambda} - T = 0.$$

Donc $\lambda = \frac{k}{T}$. □

Exemple 1. Dans le code de simulation d'un processus de Poisson de la partie 3.1, on peut comparer la valeur de l'estimateur à la valeur réelle de l'intensité. La figure 3 montre que pour un certain échantillon de taille 40, l'estimation donne la valeur 1.77 qui est de l'ordre de grandeur de 2.

6 Processus de Hawkes

Dans la suite $(N(t), t \geq 0)$ est un processus de comptage

Définition 4 (Passé d'un processus). Soit $(T_k)_{k \geq 1}$ la suite des variables temps d'arrivée du processus N et t_1, t_2, \dots des réalisations observées de ces variables. Considérons la tribu engendrée par les événements $(T_k = t_k, t_k < t)_{k \geq 1}$ qu'on note $H(t)$, alors $H(\cdot)$ est une filtration qui caractérise le passé du processus N jusqu'à la date t .

Définition 5 (L'intensité conditionnelle). Soit $(H(t), t \geq 0)$ la filtration définie ci-dessus. On appelle intensité conditionnelle du processus N la fonction λ^* définie par :

$$\lambda^*(t) = \lim_{h \rightarrow 0} \frac{\mathbb{E}(N(t+h) - N(t) | H(t))}{h}. \quad (6)$$

Cette fonction donne le risque conditionnel de l'occurrence d'un événement au temps t sachant les réalisations du processus sur l'intervalle $[0, t)$.

Théorème 4. *L'intensité conditionnelle d'un processus de comptage détermine de façon unique la structure de probabilité du processus.*

Démonstration. Voir le livre de Daley et Vere-Jones [3], page 233. \square

Définition 6 (Processus de Hawkes). Un *processus de Hawkes* $(N(t), t > 0)$ est un processus de comptage, défini par son intensité conditionnelle $\lambda^*(t)$:

$$\lambda^*(t) = \lambda + \sum_{i=1}^n \mu(t - t_i),$$

où $(t_i)_{1 \leq i \leq n}$ est la suite des dates d'occurrence jusqu'à t , λ est un réel qui représente l'intensité de base, et $\mu : [0, +\infty[\rightarrow [0, +\infty[$ est la fonction d'excitation.

On rappelle que :

$$\begin{aligned} \forall t, h > 0, \quad \mathbb{P}(N(t+h) - N(t) = 0 | H(t)) &= 1 - \lambda^*(t)h + o(h), \\ \mathbb{P}(N(t+h) - N(t) = n | H(t)) &= o(h), \quad \forall n > 1, \\ \mathbb{P}(N(t+h) - N(t) = 1 | H(t)) &= \lambda^*(t)h + o(h). \end{aligned}$$

Dans ce cas l'intensité du processus dépend non seulement du temps mais aussi des événements précédents. L'influence de ces événements sur l'intensité est donnée par le terme de droite.

Remarque 2. — μ s'appelle fonction d'excitation et donne la contribution des événements arrivés à la fonction d'intensité λ^* .

— Si on choisit μ décroissante alors l'influence des événements récents est plus importante de celle des anciens événements.

— Un choix commun de μ est la fonction exponentielle

$$\mu(t) = \alpha \exp(-\beta t).$$

Dans ce cas chaque événement augmente instantanément λ^* par α . L'influence de cet événement décroît avec le temps au taux β . Pour que le processus soit bien défini il faut que $\alpha < \beta$ (conséquence de la proposition 5 ci-dessous).

— Un autre choix commun de la fonction μ est la fonction puissance[10]

$$\mu(t) = \frac{k}{(c+t)^p}$$

avec k, c et p des nombres positifs. Ce type est plus adéquat quand on s'intéresse aux processus marqués en sismologie, définis dans la remarque 4. Dans ce cas, cette fonction correspond à la formule d'Omori pour la fréquence des répliques des tremblements de terre [15], page 10.

— On peut écrire l'intensité conditionnelle d'un processus de Hawkes sous la forme suivante :

$$\lambda^*(t) = \lambda + \int_0^t \mu(t-s) dN(s).$$

Proposition 5. *Pour que le processus de Hawkes soit bien défini il faut que :*
 $n = \int_0^\infty \mu(s) ds < 1$

Démonstration. On reprend la preuve de l'article [8].

Considérons $g(t) = \mathbb{E}(\lambda^*(t))$. Alors

$$g(t) = \mathbb{E}(\lambda + \int_0^t \mu(t-s) dN(s)) = \lambda + \int_0^t \mu(t-s) \mathbb{E}(dN(s))$$

avec

$$\lambda^*(s) = \lim_{h \rightarrow 0} \frac{\mathbb{E}(N(s+h) - N(s) | H(s))}{h} = \frac{\mathbb{E}(dN(s) | H(s))}{ds}.$$

Donc

$$g(s) = \mathbb{E}(\lambda^*(s)) = \frac{\mathbb{E}(\mathbb{E}(dN(s) | H(s)))}{ds} = \frac{\mathbb{E}(dN(s))}{ds}.$$

D'où

$$\mathbb{E}(dN(s)) = g(s) ds.$$

Par suite

$$g(t) = \lambda + \int_0^t \mu(t-s) g(s) ds = \lambda + \int_0^t g(t-s) \mu(s) ds.$$

Pour résoudre cette équation, Asmussen[1] distingue le cas : le cas $n < 1$, du cas $n \geq 1$. Asmussen montre dans la Proposition 7.4 que si $n < 1$:

$$g(t) = \mathbb{E}(\lambda^*(t)) \longrightarrow \frac{\lambda}{1-n}.$$

Alors que dans le cas contraire $\lambda^*(t)$ diverge rapidement et le processus $N(\cdot)$ explose. \square

Remarque 3. Les processus de Hawkes comme processus de branchement :

L'intensité du processus s'écrit :

$$\lambda^*(t) = \lambda + \sum_{i=1}^n \mu(t - t_i).$$

On peut voir ce processus comme un processus de branchement. En effet, il est composé d'un ensemble d'événements qui arrivent selon un processus de Poisson homogène d'intensité λ , on les appelle des "immigrants". Chaque immigrant arrivé à une date t_i , peut engendrer d'autres événements qu'on appelle des "descendants", à des instants $t \geq t_i$, au taux $\mu(t - t_i)$. Ces descendants peuvent aussi engendrer de nouveaux descendants pour constituer de nouvelles générations et ainsi de suite. Le nombre total des descendants de chaque immigrant suit un processus de Poisson de moyenne $n = \int_0^\infty \mu(t) dt$.

On s'intéresse dans la section 8 à l'estimation des paramètres du modèle à partir des événements arrivés. Zhuang, Ogata et Vere-Jones(2002) [17] se sont intéressés à l'estimation de l'intensité λ du processus de base à partir des événements qui sont des immigrants, ce processus peut avoir des interprétations physiques importantes. Ainsi, il faut une démarche qui permet de distinguer les immigrants des descendants, c'est ce qu'on appelle en anglais *Stochastic declustering*.

Si $n \geq 1$, alors ceci signifie que chaque événement qui arrive génère au moins un autre événement, donc en moyenne chaque événement va générer une infinité de descendants, ce qui explique l'explosion du processus.

Remarque 4. Les processus marqués

Quand les événements du processus possèdent une caractéristique observable, on peut l'introduire dans le modèle comme variable supplémentaire qu'on appelle la marque du processus. Ainsi, on parle de processus marqués $\{(t_i, m_i)\}$ sur $[0, T) \times M$ où M est l'espace des marques.

En sismologie, la magnitude des séismes peut être intégré comme une marque. En effet, la distribution des répliques peut dépendre de ces magnitudes. Ainsi on observe des événements $\{(t_i, m_i)\}$ avec $m_i \geq M_0$ où M_0 est la magnitude seuil. L'intensité conditionnelle de ces processus marqués peut s'écrire, d'après Ogata [10], sous la forme :

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} c(m_k) \mu(t - t_k)$$

avec

$$c(m_k) = \exp(b(m_k - M_0)).$$

Le coefficient b mesure l'effet de la magnitude sur la génération de nouvelles répliques, et μ est la fonction d'excitation définie auparavant.

On peut prendre $\mu(t) = \alpha \exp(-\beta t)$ ou $\mu(t) = \frac{k}{(t+c)^p}$.

On dit que les marques d'un processus sont indépendantes, si en connaissant les dates des événements $\{t_i\}$, les marques sont mutuellement indépendantes, et

la distribution de la marque m_i dépend uniquement de t_i . C'est le cas quand les m_i sont identiquement distribuées et indépendantes.

On dit que les marques sont imprévisibles, si m_i est indépendante de toutes les dates et les marques $\{t_j, m_j\}$ des événements précédents $t_j < t_i$. Le cas le plus commun de processus avec des marques imprévisibles est celui où elles sont, conditionnellement au passé du processus, identiquement distribuées et indépendantes.

Pour un processus marqué, on parle de processus de base qui correspond au processus des dates des événements sans les marques. On note l'intensité conditionnelle de ce processus $\lambda_g(t)$. Celle-ci peut dépendre du passé des marques ainsi que du passé des temps d'arrivée. Ainsi l'intensité conditionnelle du processus marqué est donnée par :

$$\lambda^*(t, m) = \lambda_g(t) f(m|t)$$

où $f(m|t)$ désigne l'intensité conditionnelle de la marque à l'instant t sachant le passé du processus jusqu'à cet instant. Daley et Vere Jones (2003) [3] donnent dans les exemples (c) et (d) de la section 6.4 plus de détails sur les processus de Hawkes avec des marques imprévisibles.

7 Simulation des processus de Hawkes

7.1 Algorithme d'amincissement classique

Dans cette section on s'intéresse à la simulation des processus de Hawkes par l'algorithme d'amincissement.

Dans la suite on considère le cas où la fonction d'excitation μ est la fonction exponentielle, $\mu(t) = \alpha \exp(-\beta t)$ et on travaille sur un intervalle $(0, T]$. Dans ce cas, l'intensité conditionnelle du processus de Hawkes est donnée par

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} \alpha \exp(-\beta(t - t_k)).$$

Connaissant t_1, t_2, \dots, t_k , λ^* est déterministe sur $[t_k, t_{k+1}]$, donc on peut voir la génération du point t_{k+1} comme la génération du premier point d'un processus de Poisson non homogène. On présente les étapes de la simulation ci-dessous, il est conseillé de se référer à l'algorithme d'amincissement d'un processus de Poisson non homogène à la section 4.2.2 :

1. Le premier point du processus est généré par un processus de Poisson homogène de paramètre λ .
2. Connaissant t_1, t_2, \dots, t_k , alors la génération du point t_{k+1} correspond à la génération du premier point d'un processus de Poisson non homogène d'intensité $\lambda^*(t) = \lambda + \sum_{k, t_k < t} \alpha \exp(-\beta(t - t_i))$ sur l'intervalle $[t_k, T]$:
 - On génère un nouveau point candidat s du processus de Poisson homogène de paramètre $\lambda_{sup} = \lambda^*(t_k)$. On a $\forall t \in [t_k, T] \lambda^*(t) \leq \lambda^*(t_k) = \lambda_{sup}$ par décroissance de la fonction d'excitation exponentielle.

— On accepte s avec la probabilité $\frac{\lambda^*(s)}{\lambda_{sup}}$.

3. On réitère l'opération jusqu'à ce qu'on atteigne T .

On peut prendre comme condition d'arrêt quand la probabilité d'avoir un point du processus homogène entre s et T est faible. Le programme ci-dessous donne le code de simulation des processus de Hawkes par amincissement sur \mathbb{R} .

```

sim.Hawkes.classic=function(a,b,lambda,T){
  t=c();s=0          #t est le vecteur des temps d'occurrence du processus
  intensity . local=function(u,t,lambda){
    # fonction d'intensité d'un processus de Hawkes locale aux algorithmes
    de simulation
  }
  #t vecteur des dates des vnements arrivés jusqu' l'instant u
  n=0
  for (d in t){
    n=n+a*exp(-b*(u-d))}
  return(lambda+n)
}
lambda.sup=intensity.local(s,t,lambda)
while (s<T){
  w=rexp(1,lambda.sup) #l'inter-arrivée
  s=s+w                #nouveau point candidat
  u=runif(1,0,1)
  lambda.temp=intensity.local(s,t,lambda)
  if (u<lambda.temp/lambda.sup){ #on accepte s avec la proba
    lambda.temp/lambda.sup
    t=c(t,s)          #on actualise le vecteur des dates d'occurrence
    lambda.sup=lambda.temp #on actualise lambda.sup lorsqu'on accepte
    le point s
  }
}
if (t[length(t)]<=T) {
  N=c(0:length(t))
  t=c(0,t)
}
else{
  p=length(t)-1
  N=c(0:p)
  t=c(0,t[1:p])
}
list (time=t,nbpoints=N)
}

#commandes d'exécution pour un exemple
set.seed(2020)

```

```

a=2;b=3;lambda=1;T=100;
res=sim.Hawkes.classic(a,b,lambda,T)
plot(res$time,res$nbpoints,xlab="time",type="s",ylab="N(t)",col="blue",
      main="Realization of Hawkes point process – classic")

```

La figure 5 donne la trajectoire du processus pour $\alpha = 0.6, \beta = 0.8$ et $\lambda = 1.2$.

7.2 Algorithme d'amincissement modifié d'Ogata

L'algorithme d'amincissement ci-dessous est similaire à l'algorithme précédent, sauf à l'étape 2. En effet, les incrémentations du processus de Poisson sont indépendantes et la génération des points s dans l'algorithme d'amincissement est séquentielle, donc à chaque fois qu'on génère un nouveau point candidat s , on peut modifier λ_{sup} pour la suite de la recherche des points par $\lambda_{sup} = \lambda^*(s)$. Contrairement à l'algorithme de simulation classique où on ne modifie λ_{sup} que si le point généré est accepté. Ceci est légitime d'après la remarque ci-dessous.

Remarque 5. Pour $0 \leq s \leq T$,

$$\begin{aligned}
\mathbb{P}(N(s, T] = 0 | N(0, s] = 0) &= \frac{\mathbb{P}(N(0, T] = 0)}{\mathbb{P}(N(0, s] = 0)} \\
&= \frac{\exp(-\int_0^T \lambda(u) du)}{\exp(-\int_0^s \lambda(u) du)} \\
&= \exp\left(-\int_s^T \lambda(u) du\right) \\
&= \mathbb{P}(N(s, T] = 0).
\end{aligned}$$

Le code ci-dessous donne le code de programmation de l'algorithme d'Ogata sur R.

```

sim.Hawkes.Ogata=function(a,b,lambda,T){
  t=c();s=0          #t est le vecteur des temps d'occurrence du processus
  intensity . local=function(u,t,lambda){
    # fonction d'intensité d'un processus de Hawkes locale aux algorithmes
    # de simulation
    #t vecteur des dates des vnements arrivs jusqu' l'instant u
    n=0
    for (d in t){
      n=n+a*exp(-b*(u-d))}
    return(lambda+n)
  }
  lambda.sup=intensity.local(s,t,lambda)
  while (s<T){
    w=rexp(1,lambda.sup) #l'inter-arrivee
    s=s+w                #nouveau point candidat
    u=runif(1,0,1)
  }
}

```

```

lambda.temp=intensity.local(s,t,lambda)
if (u<lambda.temp/lambda.sup){ #on accepte s avec la proba
  lambda.temp/lambda.sup
  t=c(t,s) #on actualise le vecteur des dates d'occurence
}
lambda.sup=lambda.temp #on modifie lambda.sup chaque itration
}
if (t[length(t)]<=T) {
  N=c(0:length(t))
  t=c(0,t)
}
else {
  p=length(t)-1
  N=c(0:p)
  t=c(0,t[1:p])
}
list (time=t,nbpoints=N)
}

#commandes d'execution pour un exemple
set.seed(2020)

a=2;b=3;lambda=1;T=100;
res=sim.Hawkes.Ogata(a,b,lambda,T)
plot(res$time,res$nbpoints,type="s",ylab="N(t)",col="blue",
      main="Realization of Hawkes point process – Ogata")

```

La figure 4 donne la trajectoire du processus pour $\alpha = 0.6, \beta = 0.8$ et $\lambda = 1.2$. Les trajectoires simulées par les deux algorithmes ont la même allure, on donnera dans la section suivante des outils pour vérifier qu'elles correspondent bien à la même loi.

7.3 Test des algorithmes de simulation des processus de Hawkes

7.3.1 Application au cas du processus de Poisson homogène

Dans cette section nous allons utiliser la méthode QQ-Plot qui consiste en la comparaison des quantiles empiriques et des quantiles théoriques.

D'abord nous testons la justesse des algorithmes de simulation d'un processus de Hawkes en les appliquant au cas particulier des processus de Poisson homogène qui correspondent au cas où $\alpha = \beta = 0$.

- Pour λ et T choisis, l'algorithme de simulation nous donne les dates de survenance du processus de Hawkes de paramètres $\lambda, \alpha = 0, \beta = 0$ et T . En faisant des différences entre les dates consécutives on obtient les inter-arrivées, il s'agit de l'échantillon empirique pour le test QQ-plot.

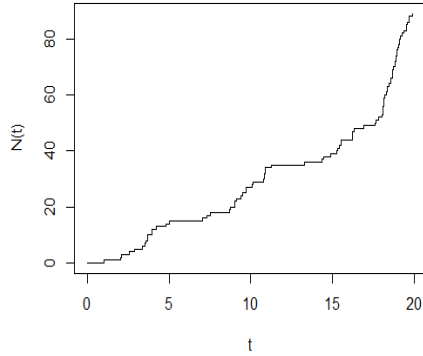


FIGURE 4 – Processus de Hawkes simulé par l'algorithme d'Ogata

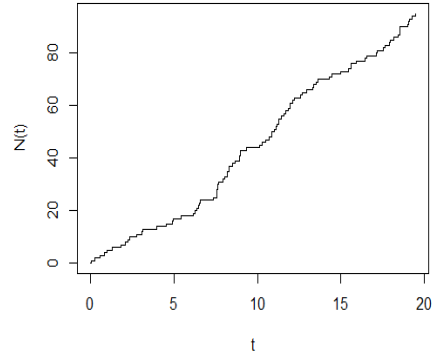


FIGURE 5 – Processus de Hawkes simulé par l'algorithme d'amincissement classique

- Ces inter-arrivées doivent suivre une loi exponentielle si le processus est bien un processus de Poisson homogène. Donc on génère une suite de variables i.i.d. qui suivent une loi exponentielle de paramètre λ , ce sera notre échantillon théorique pour le test QQ-plot.

On constate dans les figures 6 et 7 que le nuage de points peut être ajusté par la droite d'équation $y = x$. Ainsi, les quantiles théoriques et empiriques sont identiques. Par suite, les processus générés sont effectivement des processus de Poisson homogènes. Le code ci-dessous donne le code pour faire ces tests sur R.

```
##Pour tester l'algorithme d'amincissement classique##
test_QQplot=function(lambda,T){
  L=sim.Hawkes.classic(3,50,lambda,T)$time
  I=c(L[1], diff (L)) #vecteur des interarrivee
  return(I)}

#Commandes d'execution:
I=test_QQplot(1,100)
x=rexp(length(I),1) #vecteur de variables exponentielles de parametre 20
qqplot(x,I,main="Q-Q Plot ",xlab="Theoretical Quantiles",ylab="Sample
quantiles(Hawkes-classic)")
abline(0,1)

##Pour tester l'algorithme d'Ogata##
test_QQplot_Ogata=function(lambda,T){
```



```

L=sim.Hawkes.Ogata(0,10,lambda,T)$time
I=c(L[1], diff(L)) #vecteur des interarrivee
return(I)}

#Commandes d'execution:
I=test_QQplot_Ogata(1,100)
x=rexp(length(I),1) #vecteur de variables exponentielles de parametre 20
qqplot(x,I,main="Q-Q Plot ",xlab="Theoretical Quantiles",ylab="Sample
quantiles(Hawkes-Ogata)")
abline(0,1)

```

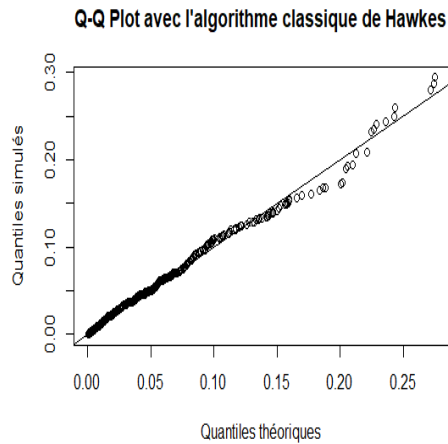


FIGURE 6 – Q-Q-Plot pour l'algorithme d'amincissement classique

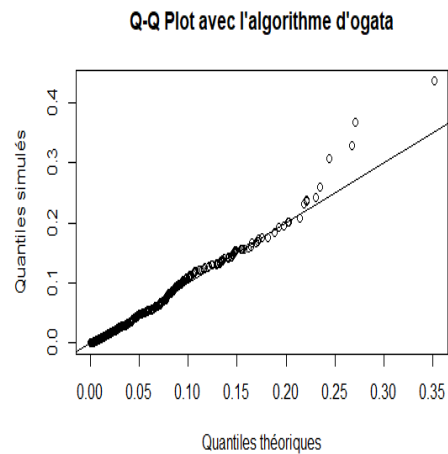


FIGURE 7 – Q-Q-Plot pour l'algorithme d'Ogata

7.3.2 Comparaison des deux algorithmes

Dans cette partie on va tester si les deux algorithmes : celui d'Ogata et l'algorithme d'amincissement classique, donnent des processus de même loi ; celle d'un processus de Hawkes. On utilise pour ceci le test d'enveloppe.

- On trace l'enveloppe des trajectoires du processus de Hawkes sur une durée T de paramètres λ, α, β définies, en utilisant l'algorithme d'Ogata. Concrètement, on trace l'enveloppe en générant un grand nombre de trajectoires.
- On superpose sur le graphe, des trajectoires du processus de Hawkes avec les mêmes paramètres simulées par l'algorithme d'amincissement classique.
- La figure 8 montre que ces trajectoires se trouvent dans l'enveloppe donc les deux algorithmes donnent des processus de même loi.

Le code ci-dessous donne le code pour faire le test d'enveloppe sur R.

```

envelope_Hawkes=function(T,n,a,b,lambda){
  plot.new()
  res=sim.Hawkes.Ogata(a,b,lambda,T)
  t=res$time
  N=res$nbpoints
  plot(t,N,type="s",col="green",ylab="N(t)")
  for (i in 1:n){
    par(new = T)
    res=sim.Hawkes.Ogata(a,b,lambda,T)
    v=res$time
    N=res$nbpoints
    plot(v,N,type="s",col="green",axes=F,xlab="",ylab="")
  }
  for (i in 1:n){
    par(new = T)
    res=sim.Hawkes.classic(a,b,lambda,T)
    v=res$time
    N=res$nbpoints
    plot(v,N,type="s",col="red",axes=F,xlab="",ylab="")
  }}
#commande d'execution pour T=100 et pour 200 echantillons
a=2;b=50;lambda=1.2
envelope_Hawkes(100,200,a,b,lambda)

```

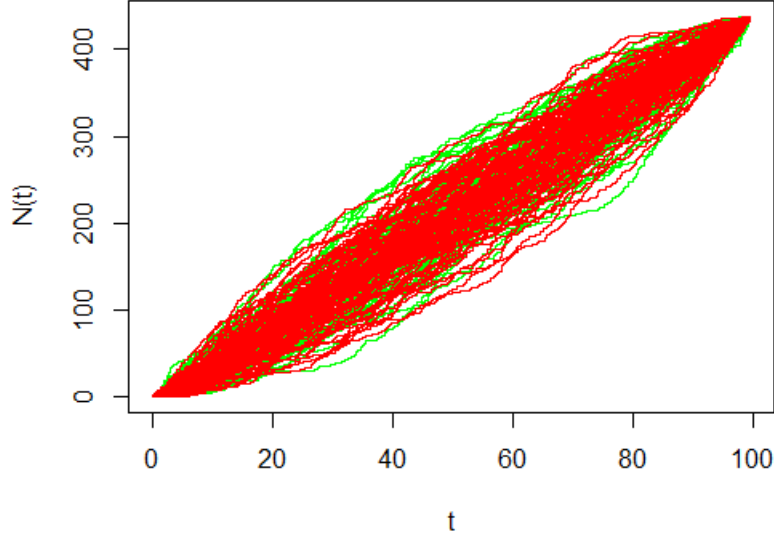


FIGURE 8 – Test d’enveloppe, les courbes en vert proviennent de l’algorithme d’Ogata, les courbes rouges de l’algorithme d’amincissement classique

8 Inférence statistique sur les processus de Hawkes

8.1 La fonction de vraisemblance

Soit $N(\cdot)$ un processus de Hawkes sur $[0, T]$ d’intensité $\lambda^*(t) = \lambda + \int_0^t \mu(t-s)dN(s)$. Dans cette partie on cherche à estimer les paramètres α , β et λ . On va utiliser la méthode de maximisation de la vraisemblance.

Daley et Vere-Jones [3][Proposition 7.2.III.] donnent le résultat suivant

Théorème 5. *Si on note t_1, \dots, t_k les réalisations du processus $N(\cdot)$ sur $[0, T]$, alors la fonction de vraisemblance L de $N(\cdot)$ est :*

$$L = \prod_{i=1}^k \lambda^*(t_i) \exp \left(- \int_0^T \lambda^*(u) du \right).$$

La log-vraisemblance de $N(\cdot)$ est

$$\ell = \sum_{i=1}^k \log(\lambda^*(t_i)) - \int_0^T \lambda^*(u) du. \quad (7)$$

Dans le cas où la fonction d'excitation est $\mu(t) = \alpha \exp(-\beta t)$, on peut simplifier l'expression de la fonction log-vraisemblance sur $[0, t_k]$:

$$\begin{aligned}
\int_0^{t_k} \lambda^*(u) du &= \int_0^{t_1} \lambda^*(u) du + \sum_{i=1}^{k-1} \int_{t_i}^{t_{i+1}} \lambda^*(u) du \\
&= \int_0^{t_1} \lambda du + \sum_{i=1}^{k-1} \left(\int_{t_i}^{t_{i+1}} \lambda + \sum_{t_j < u} \alpha \exp(-\beta(u - t_j)) du \right) \\
&= \lambda t_k + \alpha \sum_{i=1}^{k-1} \int_{t_i}^{t_{i+1}} \sum_{j=1}^i \exp(-\beta(u - t_j)) du \\
&= \lambda t_k + \alpha \sum_{i=1}^{k-1} \sum_{j=1}^i \int_{t_i}^{t_{i+1}} \exp(-\beta(u - t_j)) du \\
&= \lambda t_k - \frac{\alpha}{\beta} \sum_{i=1}^{k-1} \sum_{j=1}^i [\exp(-\beta(t_{i+1} - t_j)) - \exp(-\beta(t_i - t_j))] \\
&= \lambda t_k - \frac{\alpha}{\beta} \sum_{i=1}^{k-1} [\exp(-\beta(t_k - t_i)) - \exp(-\beta(t_i - t_i))] \\
&= \lambda t_k - \frac{\alpha}{\beta} \sum_{i=1}^{k-1} [\exp(-\beta(t_k - t_i)) - 1]. \quad (8)
\end{aligned}$$

Donc

$$\ell = \sum_{i=1}^k \ln(\lambda + \alpha \sum_{j=1}^{i-1} \exp(-\beta(t_i - t_j))) - \lambda t_k + \frac{\alpha}{\beta} \sum_{i=1}^k [\exp(-\beta(t_k - t_i))].$$

Pour réduire la complexité du calcul du premier terme, on introduit, pour $i = 1, 2, \dots, k$,

$$A(i) = \sum_{j=1}^{i-1} \exp(-\beta(t_i - t_j)).$$

Les $A(i)$ se calculent récursivement par

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

Aussi

$$\ell = \sum_{i=1}^k \ln(\lambda + \alpha A(i)) - \lambda t_k + \frac{\alpha}{\beta} \sum_{i=1}^k [\exp(-\beta(t_k - t_i)) - 1]. \quad (9)$$

Le code ci-dessous donne le code de calcul de l'opposé la log-vraisemblance d'un échantillon de Hawkes simulé par l'algorithme d'amincissement sur R, il est de complexité linéaire :

```
log_likelihoood=function(x,t){  #t l'echantillon
  lambda=x[1];alpha=x[2];beta=x[3]
  if (x[3]<x[2]){return(0)}
  n=length(t)
  a1=rep(0,n)
  for (i in 2:n){
    a1[i]=exp(-(t[i]-t[i-1])*beta)*(a1[i-1]+1)  #A(i)
  }
  l= -t[n]*lambda +
    sum(exp(-beta*(t[n]-t))-1)*alpha/beta+sum(log(lambda+alpha*a1))
  return(-l)}

```

Ici le code nous renvoie l'opposé de la log-vraisemblance car nous allons nous intéresser à un problème de minimisation dans la suite. Et comme les processus de Hawkes ne sont définis que si $\alpha < \beta$, on impose à la log-vraisemblance d'être nulle quand cette condition n'est pas vérifiée.

Nous avons tracé, pour un processus de Hawkes de paramètres $\alpha = 0.6$, $\beta = 0.8$ et $\lambda = 1.2$, sur l'intervalle de temps $T = 60$, l'opposé de la log-vraisemblance en fonction de α et β qui varient entre 0.1 et 4 dans la figure 9. Nous observons que dans ce cas la fonction est convexe. Afin de d'observer le comportement de la log-vraisemblance de plus près, nous l'avons tracé sur la figure 10 pour α et β qui varient cette fois entre 0.1 et 1. Nous observons mieux que les vrais paramètres $\alpha = 0.6$ et $\beta = 0.8$ sont proches du minimum de la fonction.

Le code R pour tracer cette log-vraisemblance est présenté ci-dessous.

```
x <- seq(0.1,1,by=0.02)  #alpha
y <- seq(0.1,1,by=0.02)  #beta

#Les vrais parametres de l'echantillon
alpha=0.4;beta=0.8;lambda=1.2;T=1000
t=sim.Hawkes.Ogata(alpha,beta,lambda,T)$time
z=matrix(rep(0, length(x)*length(y)), length(x), length(y))
for(i in 1:length(x)){
  for(j in 1:length(y)){
    z[i,j]=log_likelihoood(c(lambda,x[i],y[j]),t)
  }
}

pmat<-persp(x,y,z, xlim = range(x), ylim = range(y),zlim = range(z), na.rm
= TRUE),xlab="alpha",ylab="beta", zlab = "-lnL",theta = 140, phi =
20,scale=TRUE, axes=T, ticktype="detailed")

#On ajoute au graphe le point qui represente les vrais paramtres de
l'echantillon
e=c(lambda,alpha,beta)

```

```

x0=alpha;y0=beta;z0=log_likelihood(e,t)
trans3d <- function(x,y,z, pmat) {
  tr <- cbind(x,y,z,1) %*% pmat
  list (x = tr [,1] /tr [,4], y= tr [,2] /tr [,4])
}
points(trans3d(x0,y0,z0,pmat),pch=16, col="red")

```

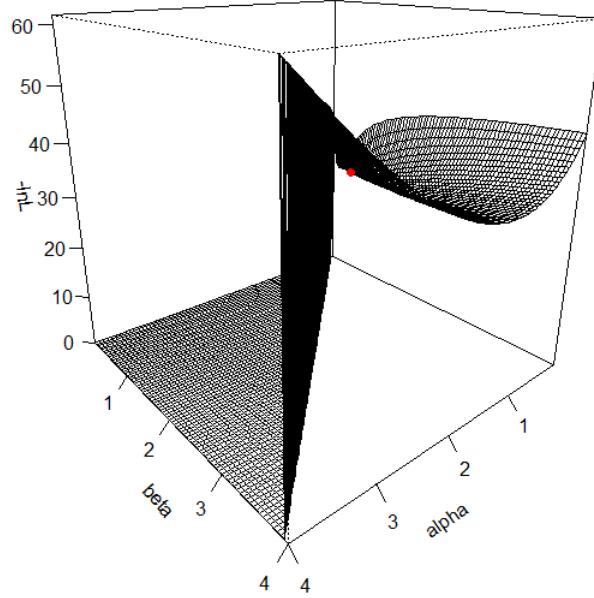


FIGURE 9 – Tracé de la fonction de vraisemblance en fonction de α et β allant de 0.1 à 4, avec $\lambda = 1.2, T = 10$. Le point en rouge correspond aux vrais paramètres de l'échantillon : $\alpha = 0.6$ et $\beta = 0.8$.

Remarque 6. Inférence sur les processus marqués

Dans le cas des processus marqués en sismologie présentés dans la remarque 4. On observe des événements $\{(t_i, m_i)\}$ avec $\{m_i\}$ les magnitudes des séismes. L'intensité conditionnelle de ces processus marqués peut s'écrire, d'après Ogata [10], sous la forme :

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} \exp(b(m_k - M_0))\mu(t - t_k)$$

avec $\mu(t) = \alpha \exp(-\beta t)$ ou $\mu(t) = \frac{k}{(t+c)^p}$.

Nous discuterons plus de ces modèles dans la section 12. Lomnitz (1966) a affirmé que la fréquence des magnitudes des séismes ne dépend pas des temps d'occurrence, ce qui signifie que les $\{m_i\}$ sont indépendants des $\{t_i\}$. Ainsi la densité jointe des magnitudes et des temps d'occurrence est le produit des deux

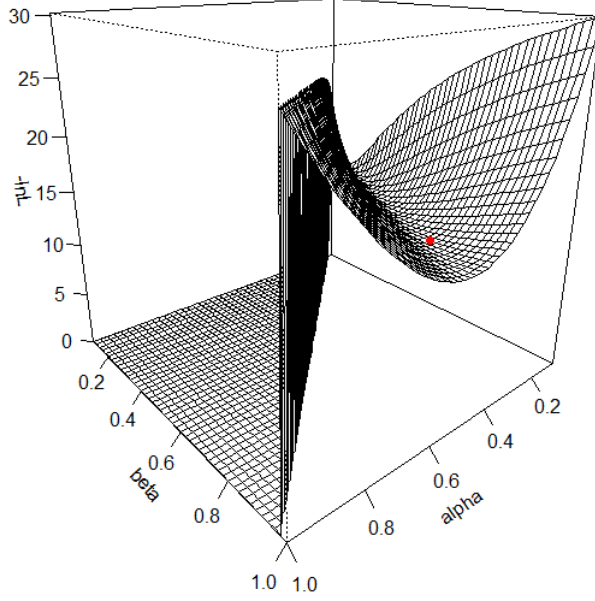


FIGURE 10 – Tracé de la fonction de vraisemblance en fonction de α et β allant de 0.1 à 1, avec $\lambda = 1.2, T = 10$. Le point en rouge correspond aux vrais paramètres de l'échantillon étudié.

densités marginales. Donc la maximisation de la log-vraisemblance revient à maximiser chacune des deux parties séparément. Et l'expression (11.3) de ℓ reste valable pour la partie des temps d'occurrence. Sans cette hypothèse d'indépendance, il faut maximiser la log-vraisemblance totale pour trouver les paramètres (b, α, β) dans le cas exponentiel, et (b, k, c, p) dans le cas de la fonction puissance, ce qui est très coûteux.

Concernant la densité des magnitudes, la loi de Gutenberg et Richter affirme que les magnitudes suivent une loi exponentielle.

$$\mathbb{P}(\text{magnitude} > M) = \exp(-aM).$$

8.2 Le gradient et la matrice hessienne

D'après la forme de la log-vraisemblance (9), on voit qu'elle est continue et différentiable, par contre on ne peut pas conclure sur sa concavité.

Ozaki [14] a donné le calcul des termes de la matrice Hessienne que nous reprenons ci-dessous. D'après l'expression de la log-vraisemblance (9), le gradient

est donné par :

$$\frac{\partial \ell}{\partial \alpha} = \sum_{i=1}^k \frac{A(i)}{\lambda + \alpha A(i)} + \frac{1}{\beta} \sum_{i=1}^k [\exp(-\beta(t_k - t_i)) - 1], \quad (10)$$

$$\frac{\partial \ell}{\partial \beta} = - \sum_{i=1}^k \alpha \frac{B(i)}{\lambda + \alpha A(i)} - \alpha \sum_{i=1}^k \left[\frac{1}{\beta} (t_k - t_i) \exp(-\beta(t_k - t_i)) + \frac{1}{\beta^2} \exp(-\beta(t_k - t_i)) \right], \quad (11)$$

avec

$$\begin{cases} B(i) = \sum_{t_j < t_i} (t_i - t_j) \exp(-\beta(t_i - t_j)), \text{ pour } i \geq 2, \\ B(1) = 0 \end{cases}$$

et

$$\frac{\partial \ell}{\partial \lambda} = -t_k + \sum_{i=1}^k \frac{1}{\lambda + \alpha A(i)}. \quad (12)$$

Les termes de la matrice hessienne sont :

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

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \beta^2} &= - \sum_{i=1}^k \left[\frac{\alpha c(i)}{\lambda + \alpha A(i)} - \left(\frac{\alpha B(i)}{\lambda + \alpha A(i)} \right)^2 \right] \\ &+ \alpha \sum_{i=1}^k \left[\frac{1}{\beta} (t_k - t_i)^2 \exp(-\beta(t_k - t_i)) + \frac{2}{\beta^2} \exp(-\beta(t_k - t_i)) + \frac{2}{\beta^3} (\exp(-\beta(t_k - t_i)) - 1) \right], \end{aligned}$$

avec

$$\begin{cases} c(i) = \sum_{j=1}^i (t_i - t_j)^2 \exp(-\beta(t_i - t_j)) \quad i \geq 2, \\ c(1) = 0, \end{cases}$$

$$\frac{\partial^2 \ell}{\partial \lambda^2} = - \sum_{i=1}^k \frac{1}{(\lambda + \alpha A(i))^2}, \quad (13)$$

$$\frac{\partial^2 \ell}{\partial \alpha \partial \lambda} = - \sum_{i=1}^k \frac{A(i)}{(\lambda + \alpha A(i))^2}, \quad (14)$$

$$\frac{\partial^2 \ell}{\partial \beta \partial \lambda} = - \sum_{i=1}^k \frac{\alpha B(i)}{(\lambda + \alpha A(i))^2}. \quad (15)$$

8.3 Estimation par maximisation de la vraisemblance

Nous cherchons à optimiser la log-vraisemblance (9). Il s'agit d'un problème d'optimisation non linéaire. Dans le cas d'une fonction d'excitation exponentielle, la log-vraisemblance est concave (Voir Daley et Vere Jones [3], exemple 7.2), donc si le maximum de la vraisemblance existe, il sera unique.

Problème général : Nous cherchons à résoudre le problème d'optimisation suivant :

$$\begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.c } c_i(x) \geq 0 \ i = 1, \dots, m \end{cases} \quad (16)$$

avec $n, m \in \mathbb{N}$ et les fonctions f et $(c_i)_{i=1, \dots, m}$ sont convexes.

Dans notre cas, nous cherchons à minimiser dans \mathbb{R}^3 l'opposé de la log-vraisemblance sous quatre contraintes, ainsi : $x = (\alpha, \beta, \lambda) \in \mathbb{R}^3$, $f = -\ell = -\log(L)$ et $c_1(x) = \alpha$, $c_2(x) = \beta + \epsilon_1$, $c_3(x) = \lambda$, $c_4(x) = \beta - \alpha + \epsilon_2$. ϵ_1, ϵ_2 sont des nombres positifs petits qu'on introduit pour passer des inégalités strictes à des inégalités larges.

Pour résoudre ce problème d'optimisation convexe, on souhaite passer à un problème d'optimisation sans contraintes. Nous utilisons la méthode de **la barrière logarithmique**. L'idée consiste à introduire dans la fonction à optimiser un terme qui la pénalise quand on s'approche de la frontière des contraintes à respecter.

On définit la fonction de barrière comme :

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^m \log(c_i(x))$$

avec $\mu > 0$ le paramètre de la barrière. Si l'une des fonctions c_i s'annule ou devient négative, le terme logarithmique dans la fonction B va entraîner sa divergence.

On s'intéresse désormais au problème d'optimisation sans contraintes suivant :

$$\min_{x \in \mathbb{R}^n} B(x, \mu). \quad (17)$$

Le théorème 5 de l'article [16], page 361, nous garantit l'équivalence entre les problèmes 22 et 17, le théorème est accompagné d'une preuve dans cet article. Nous énonçons ci-dessous une version réduite de ce théorème avec les points clés pour notre optimisation.

Théorème 6. Notons $C = \{x/c_i(x) > 0, i = 1, \dots, m\}$ et soit μ_k une suite de paramètres de barrière décroissante tel que $\lim_{k \rightarrow \infty} \mu_k = 0$, si $C \neq \emptyset$ et l'ensemble des minimiseurs locaux du problème avec contraintes est non vide et borné alors :

- $B(x, \mu_k)$ est convexe sur C .
- B a un minimiseur sur C pour tout $\mu_k > 0$.
- il existe une sous-suite convergente de minimiseurs $\{x_k\}$ de $B(x, \mu_k)$ et $\lim_{k \rightarrow \infty} f(x_k) = f^*$, avec f^* est la valeur optimale de f dans le problème 22.

$$— \lim_{k \rightarrow \infty} B(x, \mu_k) = f^*$$

Ainsi en utilisant cette méthode de pénalisation on peut résoudre notre problème avec une fonction d'optimisation sans contraintes comme la méthode de Newton. C'est ce que fait la fonction *ConstrOptim* de R qui passe à un problème sans contraintes grâce à la barrière logarithmique puis minimise le problème en utilisant la fonction d'optimisation sans contraintes *Optim* de R. Dans la section suivante nous présentons les résultats de notre estimation de paramètres sur des échantillons simulés.

Estimation des paramètres α, β, λ Nous allons utiliser la fonction d'optimisation sous contraintes linéaires *ConstrOptim* de R, le code R est donné ci-dessous.

```
estimation=function(temps){ #temps est le vecteur des dates d'occurrence du
  processus qui nous interesse
  #la log-vraisemblance comme fonction d'un seul paramtre
  lnL=function(x){
    return(log_likelihood(x,temps))
  }
  c=constrOptim(theta=c(0.1,0.01,10),f=lnL,grad=NULL,ui=cbind(c(1,0,0,0)
    ,c(0,1,0,-1),c(0,0,1,1) ), ci=c(0,0,0,0.1) )
  return(c$par)
}
#Commandes d'excution
a=0.4;b=0.8;lambda=1.2;T=1000
t=sim.Hawkes.Ogata(a,b,lambda,T)$time
#lambda,alpha beta
estimation(t)
```

Pour vérifier l'efficacité de l'estimation on génère n échantillons de Hawkes simulés par l'algorithme d'amincissement de paramètres $\lambda = 1.2, \alpha = 0.6$ et $\beta = 0.8$ et on calcule les moyennes des paramètres estimés pour des valeurs différentes de T . Le code R est donné ci-dessous.

```
test_estimation=function(n,T){
  a=0.6;b=0.8;lambda=1.2;
  s=c(0,0,0)
  for (i in 1:n){
    t=sim.Hawkes.Ogata(a,b,lambda,T)$time
    s=s+estimation(t)
  }
  return(s/n)
}
#Commande d'excution
```

test_estimation(200,100)

Les tableaux ci-dessous montrent l'influence de n et T sur les résultats de l'estimation.

Effet du nombre d'échantillons n , pour $T = 100$:

	λ	α	β
$n=100$	1.53	0.60	0.91
$n=200$	1.47	0.58	0.87
$n=300$	1.50	0.60	0.96

Effet de la durée T , pour $n = 100$:

	λ	α	β
$T=100$	1.49	0.55	0.82
$T=200$	1.33	0.59	0.84
$T=300$	1.29	0.60	0.82
$T=500$	1.28	0.59	0.82
$T=1000$	1.23	0.58	0.79

On remarque que les valeurs estimées convergent vers les vrais paramètres quand on augmente le temps T , alors que ce n'est pas le cas quand on augmente le nombre d'échantillons. On peut justifier ceci par le fait que plus la durée T est grande plus on a d'informations sur le processus ce qui donne une meilleur estimation.

Pour $T = 200$ et $n = 100$, nous avons tracé dans les figures 11, 12 et 13 les histogrammes des valeurs estimées pour λ , α et β respectivement. On remarque que les valeurs estimées sont centrées à peu près autour des vrais paramètres, et ce sont ces paramètres qui ont la plus grande fréquence.

9 Validation du modèle

Pour valider un modèle de processus de Hawkes et vérifier sa pertinence pour la représentation des données, nous effectuons une analyse résiduelle présentée dans le théorème ci-dessous. On définit d'abord l'intensité cumulée par :

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

Théorème 7. Soient $\{t_1, t_2, \dots\}$ les dates d'occurrences d'un processus ponctuel N , la suite des résidus $\{\tau_1, \tau_2, \dots\} = \{\Lambda(t_1), \Lambda(t_2), \dots\}$ est la réalisation d'un processus de Poisson homogène d'intensité 1 si et seulement si le processus N est un processus de Hawkes d'intensité λ^* .

Démonstration. Voir le théorème 7.4.I de Daley et Vere-Jones [3]. □

Ainsi après avoir estimé les paramètres du modèle $(\hat{\lambda}, \hat{\alpha}, \hat{\beta})$, on calcule l'intensité cumulée estimée $\hat{\Lambda}$, et on vérifie si la suite des résidus $\{\tau_1, \tau_2, \dots\} =$

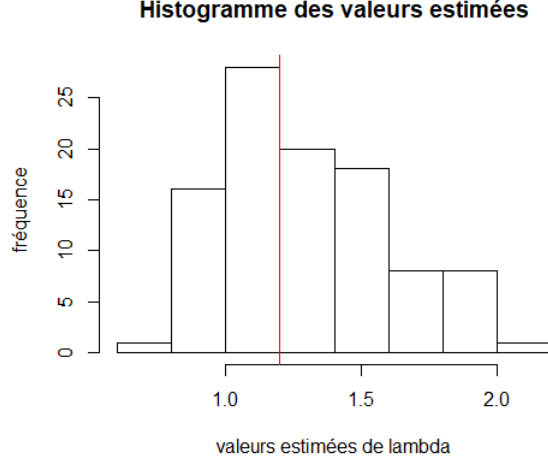


FIGURE 11 – Histogramme des valeurs estimées de λ pour 100 échantillons sur une durée $T = 200$, le vrai paramètre de l'échantillon est $\lambda = 1.2$

$\{\hat{\Lambda}(t_1), \hat{\Lambda}(t_2), \dots\}$ correspond à la réalisation d'un processus de Poisson homogène d'intensité 1, ce qui peut être fait notamment avec un test QQplot.

Remarque 7. — Dans le cas où on dispose de plusieurs modèles de processus de Hawkes, notamment avec des fonctions d'excitation μ différentes, on peut les comparer en utilisant le critère d'information Akaike (AIC). Le meilleur modèle est celui avec la plus petite valeur de AIC[10].

$\text{AIC} = \max(\log \text{vraisemblance}) - \text{vraisemblance} + 2 \times \text{nombre de paramètres}$

— Dans le cas d'une fonction d'excitation exponentielle, calculons l'intensité cumulée à un instant t avec t_k la date du dernier événement arrivé avant t ,

$$\Lambda(t) = \int_0^{t_k} \lambda^*(u) du + \int_{t_k}^t \lambda^*(u) du,$$

D'après l'équation (8) :

$$\int_0^{t_k} \lambda^*(u) du = \lambda t_k - \frac{\alpha}{\beta} \sum_{i=1}^{k-1} [\exp(-\beta(t_k - t_i)) - 1].$$

Et on a

$$\begin{aligned} \int_{t_k}^t \lambda^*(u) du &= \int_{t_k}^t (\lambda + \alpha \sum_{t_i < u} \exp(-\beta(u - t_i))) du \\ &= \lambda t - \lambda t_k - \frac{\alpha}{\beta} \sum_{i=1}^k [\exp(-\beta(t - t_i)) - \exp(-\beta(t_k - t_i))]. \end{aligned}$$

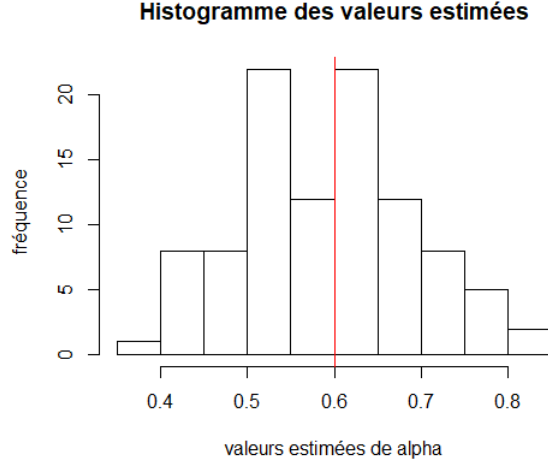


FIGURE 12 – Histogramme des valeurs estimées de α , pour 100 échantillons sur une durée $T = 200$, le vrai paramètre de l'échantillon est $\alpha = 0.6$

Donc

$$\int_{t_k}^t \lambda^*(u) du = \lambda t - \lambda t_k - \frac{\alpha}{\beta} \sum_{i=1}^{k-1} [\exp(-\beta(t - t_i)) - \exp(-\beta(t_k - t_i))] - \frac{\alpha}{\beta} [\exp(-\beta(t - t_k)) - 1].$$

Donc l'intensité cumulée s'écrit :

$$\Lambda(t) = \lambda t - \frac{\alpha}{\beta} \sum_{i=1}^k [\exp(-\beta(t - t_i)) - 1] = \lambda t - \frac{\alpha}{\beta} \sum_{t_i < t} [\exp(-\beta(t - t_i)) - 1]. \quad (18)$$

En pratique, pour vérifier que les résidus forment bien un processus de Poisson homogène d'intensité 1, on peut faire le test d'enveloppe en traçant l'enveloppe des processus de Poisson et en superposant dans le graphe les trajectoires de plusieurs échantillons des résidus. Cependant cette méthode n'est pas appliquée directement au processus qui nous intéresse mais elle vérifie plutôt la pertinence de ses paramètres. Un test statistique qu'on peut appliquer au processus, est le test de Kolmogorov-Smirnov. Dans ce test, on vérifie que la fonction de répartition des différences entre les résidus est bien celle d'une loi exponentielle de paramètre 1. L'hypothèse nulle de ce test est celle de l'égalité entre les fonctions de répartition.

Test d'enveloppe

Le code ci-dessous donne le programme R pour faire une analyse résiduelle en utilisant le test d'enveloppe, on compare l'enveloppe des résidus de process

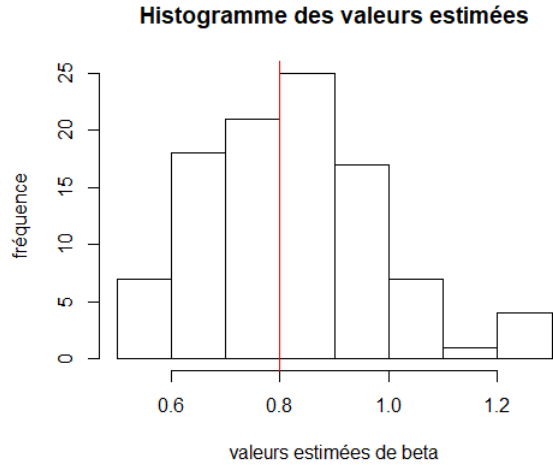


FIGURE 13 – Histogramme des valeurs estimées de β , pour 100 échantillons sur une durée $T = 200$, le vrai paramètre de l'échantillon est $\beta = 0.8$

de Hawkes simulés par l'algorithme d'Ogata, à des échantillons de processus de Poisson homogène de paramètre 1. La figure 14 montre que ces résidus forment effectivement un processus de Poisson de paramètre 1.

```

#l'intensite cumulee au point s d'un processus de Hawkes temps
int _cumul=function(s,lambda,a,b,temps){
  A=temps[temps<s]
  return(lambda*s-(a/b)*sum(exp(-b*(s-A))-1)) }

#Fonction pour calculer les residus
residu=function(t,lambda,a,b){
  res=rep(0,length(t))
  for (i in 1:length(res)){
    t.temp=t[i]
    res[i]=int _cumul(t.temp,lambda,a,b,t)
  }
  return(res)}
envelope _residuals=function(T,n,a,b,lambda){
  plot.new()
  process=sim.Hawkes.Ogata(a,b,lambda,T)
  t=process$time
  N=process$nbpoints
  plot(t,N,type="s",col="green",ylab="N(t)")
  for (i in 1:n){      #on trace les processus de Poisson(1)
    par(new = T)
  }
}

```

```

v=Poisson_homogene(T,1)
N=c(0:(length(v)-1))
plot(v,N,type="s",col="green",axes=F,xlab="",ylab="")
}
for (i in 1:n){      #on trace les residus des processs de Hawkes simules
  par(new = T)
  process=sim.Hawkes.Ogata(a,b,lambda,T)
  v=process$time
  res=residu(v,lambda,a,b)
  N=process$nbpoints
  plot(res,N,type="s",col="red",axes=F,xlab="",ylab="")
}}

#Commandes d'excution

a=0.6;b=0.8;lambda=1;T=100;
envelope_residuals(100,100,a,b,lambda)

```

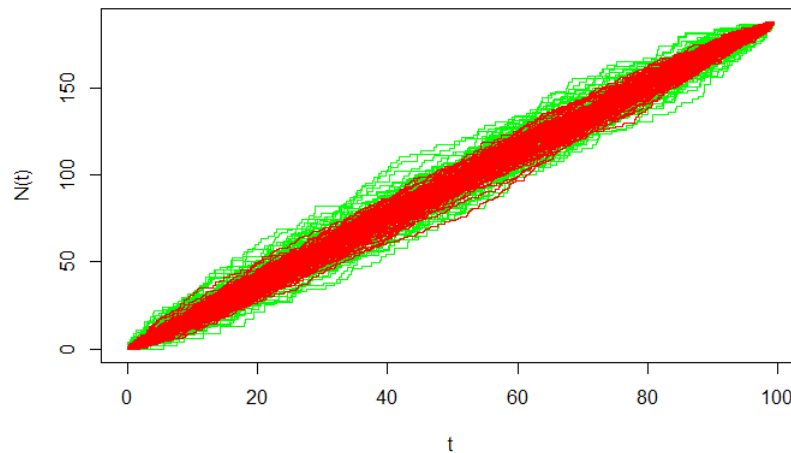


FIGURE 14 – Test d'enveloppe pour l'analyse des résidus du modèle de Hawkes avec une fonction d'excitation exponentielle. Les courbes vertes correspondent à des processus de Poisson d'intensité 1 et les courbes rouges correspondent aux résidus des processus de Hawkes qu'on a simulé

Test de Kolmogorov-Smirnov Le code ci-dessous donne le programme R pour faire une analyse résiduelle en utilisant le test de Kolmogorov-Smirnov, dans ce cas on utilise un échantillon simulé par l'algorithme d'Ogata.

```
#test de Kolmogorov sur un chantillon de Hawkes simul
```

```

a=0.6;b=0.8;lambda=3;T=100
process=sim.Hawkes.Ogata(a,b,lambda,T)$time
res_hawkes_sim=residu(process,lambda,a,b)
diff_sim=c(res_hawkes_sim[1],diff(res_hawkes_sim))#les interarrivees

ks.test(diff_sim,"pexp",1)

```

La valeur de la p -valeur du test, permet de décider si on garde notre hypothèse de résidus qui suivent un processus de Poisson à taux unitaire.

Remarque 8. Pour vérifier le modèle, on peut aussi analyser les résidus de martingales R , qui représentent l'erreur entre le modèle et les données cumulée au cours du temps. Soit N la variable de comptage du processus :

$$R(t) = N(t) - \int_0^t \lambda^*(u) du.$$

On crée une grille z_1, z_2, \dots de points entre 0 et T . Comme $R(t)$ est une martingale, alors $\mathbb{E}(R(t)) = 0$. En se basant sur cet argument, on calcule $R(z_1), R(z_2) - R(z_1), R(z_3) - R(z_2), \dots$ et il faut qu'on trouve des valeurs proches de 0.

Remarque 9. Pour plus de précision, on va construire un intervalle de confiance pour ces résidus. Notons $(Z_i)_i$ ces résidus, on a

$$\begin{aligned}
Z_i &= R(t_i) - R(t_{i-1}) = N(t_i) - \Lambda(t_i) - N(t_{i-1}) + \Lambda(t_{i-1}) \\
&= \Lambda(t_{i-1}) - \Lambda(t_i) + 1 \\
&= \tau_{i-1} - \tau_i + 1,
\end{aligned}$$

posons $X_i = -Z_i$ et $Y_i = \tau_i - \tau_{i-1}$ alors

$$X_i = Y_i - 1,$$

Y_i suit une loi exponentielle de paramètre 1, comme conséquence du théorème 7 de l'analyse résiduelle. On évalue la somme des valeurs des résidus

$$\sum_{i=1}^N X_i = \sum_{i=1}^N Y_i - N,$$

avec N le nombre d'échantillons.

Pour trouver un intervalle de confiance au risque 5%, on cherche α tel que

$$\mathbb{P} \left[\sum_{i=1}^N X_i > \alpha \right] = 0.05,$$

c-à-d

$$\mathbb{P} \left[\sum_{i=1}^N Y_i > N + \alpha \right] = 0.05, \quad (19)$$

la variable $\sum_{i=1}^N Y_i$ est la somme de N variables exponentielles de paramètre 1, donc elle suit une loi Gamma de paramètre N .

On verra dans la suite qu'on aura $N = 400$ pour des données réelles et $N = 100$ pour des données simulées. Calculons les intervalles de confiance pour ces deux valeurs.

Pour $N = 400$, on trouve que le quantile d'ordre 0.05 de la loi Gamma est $q = 433.45$. Comme

$$N + \alpha = q,$$

alors

$$\alpha = q - N = 33.45,$$

De même, pour le quantile inférieur, on cherche β tel que

$$\mathbb{P} \left[\sum_{i=1}^N X_i < \beta \right] = 0.05,$$

c-à-d

$$\mathbb{P} \left[\sum_{i=1}^N Y_i < N + \beta \right] = 0.05,$$

ce qui correspond au quantile d'ordre 0.95 de la loi Gamma, on trouve $q_{inf} = 367.68$, donc

$$\beta = q_{inf} - N = -32.32.$$

Par conséquent, comme les résidus Z_i sont l'opposé des X_i , alors les quantiles supérieur et inférieur de la somme des résidus au risque 5% pour 400 échantillons sont

$$\begin{aligned} \alpha_{sup}(N = 400) &= 32.32 \\ \alpha_{inf}(N = 400) &= -33.45. \end{aligned}$$

Pour $N = 100$, on trouve de la même façon que $q = 84,14$ et $q_{inf} = 116,99$ donc

$$\begin{aligned} \alpha &= 100 - 84,14 = 15.86 \\ \beta &= 100 - 116,99 = -16.99, \end{aligned}$$

par conséquent les quantiles des résidus sont :

$$\begin{aligned} \alpha_{sup}(N = 100) &= 16.99 \\ \alpha_{inf}(N = 100) &= -15.86 \end{aligned}$$

En général, en suivant cette approche pour chaque nombre d'événements on peut construire une bande de confiance. On superpose la martingale $\sum Y_i$ aux courbes des quantiles de la loi Gamma pour chaque nombre d'événements. Le code ci-dessous donne le programme R pour construire cette bande à partir des dates d'occurrence et du vecteur des résidus.

```

conf.band<-function(t,intcum){
  #intcum vecteur des residus=image de t par int cumule
  #diff(intcum) les rsidus Yi
  mart=cumsum(c(0,diff(intcum)))
  nb=c(1:length(t))
  #fonction qui calcule le seuil
  seuil<-function(N){
    return(qgamma(0.05,shape=N,scale=1,lower.tail = F))
  }
  seuil_low<-function(N){
    return(qgamma(0.95,shape=N,scale=1,lower.tail = F))
  }
  s_up<-Vectorize(seuil,vectorize.args = "N")
  s_low<-Vectorize(seuil_low,vectorize.args = "N")
  #seuil suprieur N par N
  up=s_up(nb)
  low=s_low(nb)
  result <- data.frame(martingale=mart,upper=up,lower=low)
  return( invisible ( result ))
}

```

10 Applications

10.1 Application à des données simulées

Afin de vérifier les résultats présentés précédemment, nous avons effectué une inférence statistique sur les données fournies par un processus de Hawkes simulé. L'estimation des paramètres par maximisation de la vraisemblance nous donne :

$$\hat{\alpha} = 0.6, \hat{\beta} = 0.94, \hat{\lambda} = 1.37.$$

Les valeurs réelles sont $\alpha = 0.6, \beta = 0.8$ et $\lambda = 1.2$, les valeurs estimées sont donc proches des valeurs réelles. On peut interpréter λ comme le nombre d'immigrants qui arrivent par unité de temps, ainsi pour une échelle journalière, 2 nouveaux événements arrivent par jour.

Le programme ci-dessous donne le code R pour tracer l'intensité conditionnelle superposée à la trajectoire du processus, pour faire le test d'enveloppe, pour tracer les résidus de martingale ainsi que pour comparer les fonctions de répartition théorique et empirique du test de Kolmogorov-Smirnov.

```

par(mfrow=c(2,2))
#processus
alpha=0.6;beta=0.8;lambda_reel=1.2;T=100;
process=sim.Hawkes.Ogata(alpha,beta,lambda_reel,T)

```

```

t=process$time
N=process$nbpoints
#estimation des paramtres
estim=estimation(t) #1.8970051 0.5304364 0.7154494
estim
lambda=estim[1];a=estim[2];b=estim[3]
#parameters
T=t[length(t)]
#lambda=2 un evenements tous les 2 jours
x=seq(0,T,by=0.01)
n=length(x)
int=rep(0,n)
for (i in 1:n){
  int[i]=intensity(lambda,a,b,x[i],t)
}
#####GRAPHE1#####
plot(x,int,type="l",xlab="days",ylab="",main="Trajectory and conditional
intensity",cex.main=0.9)
legend("bottomright",legend=c("Trajectory of events", "conditional
intensity"),col=c("red", "black"), lty=1,1, cex=0.65)
par(new = T)
plot(t,N,type="s",col="red",axes=F,xlab="",ylab="")
#test d'enveloppe
#####GRAPHE2#####
envelope_residuals(100,100,a,b,lambda)

#Kolmogorov-Sminrov
res=residu(t,lambda,a,b)
diff_sim=c(res[1],diff(res))
p=ks.test(diff_sim,"pexp",1)
D=p$statistic
#####GRAPHE3#####
inter=sort(diff_sim)
plot(ecdf(inter),ylab="",xlab=expression(paste(Lambda,"(days)")),main =
'Cumulative Distribution functions',cex.main=0.9,pch=20)
legend("bottomright",legend=c("empirical function", "theoretical
function"),col=c("black", "red"), lty=1,1, cex=0.65)
mtext(paste("KS Distance=", round(D,2)),side=3,cex=0.75)
#par(new=T)
lines(inter, pexp(inter,1),ylab="",col="red",type="l")

# rsidus de martingale
# crons une grille z partir de t
n=200
ind=seq(1,length(t),by=floor(length(t)/n))

```

```

z=rep(0,length(ind))
for (i in 1:length(z)){
  z[i]=t[ind[i]]
}
residus__mart=function(z,t,N,lambda,a,b){
  #z la grille
  #N le nombre de points correspondant au processus t
  res=rep(0,length(z))
  for (i in 2:length(res)){
    t_actu=max(t[t<z[i]]) #dernier vnement arriv avant L[i]
    k=which(t==t_actu)
    res[i]=N[k]-int__cumul(z[i],lambda,a,b,t) #rsidu l'instant L[i]
  }
  return(res)
}
#commande d'execution
res_z=residus__mart(z,t,N,lambda,a,b)
#liste des differences
diff_z=c(res_z[1],diff(res_z))
#####GRAPHE4#####
plot(z, diff_z,xlab="days",ylab="residuals",main = 'Martingale
residuals',cex.main=0.9,type='l',col=2)

```

La figure 15 fait une présentation synthétique des procédures d'inférence implémentées dans le code ci-dessus.

Le test d'enveloppe est constitué des résidus de 100 processus de Hawkes simulés avec les paramètres $\hat{\alpha}$, $\hat{\beta}$ et $\hat{\lambda}$ dans l'intervalle de temps $[0, T]$ avec $T = 100$ jours. Ces résidus se trouvent tous dans l'enveloppe, ce qui confirme qu'ils forment des processus de Poisson à taux unitaire. Les résidus de martingale oscillent autour de 0, par conséquent l'erreur entre le modèle et les données accumulées au fil du temps est faible.

Concernant le test de Kolmogorov-Smirnov des inter-arrivées des résidus. Le test renvoie une valeur p égale à 0,65. Cette valeur étant élevée, nous ne pouvons pas rejeter l'hypothèse d'avoir une séquence qui suit une loi exponentielle de taux unitaire. La courbe de test de Kolmogorov-Smirnov montre la fonction de répartition des données empiriques et celle de la distribution exponentielle avec un taux unitaire. Les deux fonctions sont très proches, et la distance de Kolmogorov-Smirnov est égale à 0.03, qui est une petite valeur. Pour les résidus de martingale, on voit qu'ils oscillent autour de 0. La figure 16 montre le tracé de la martingale $\sum Y_i$ de l'équation 19 et de sa bande de confiance. On voit qu'elle est bien dans la bande. Par conséquent, tous ces tests ne rejettent pas l'hypothèse nulle, selon laquelle les résidus forment un processus de Poisson à taux unitaire, de sorte que le processus est un processus de Hawkes de paramètres α , β et λ .

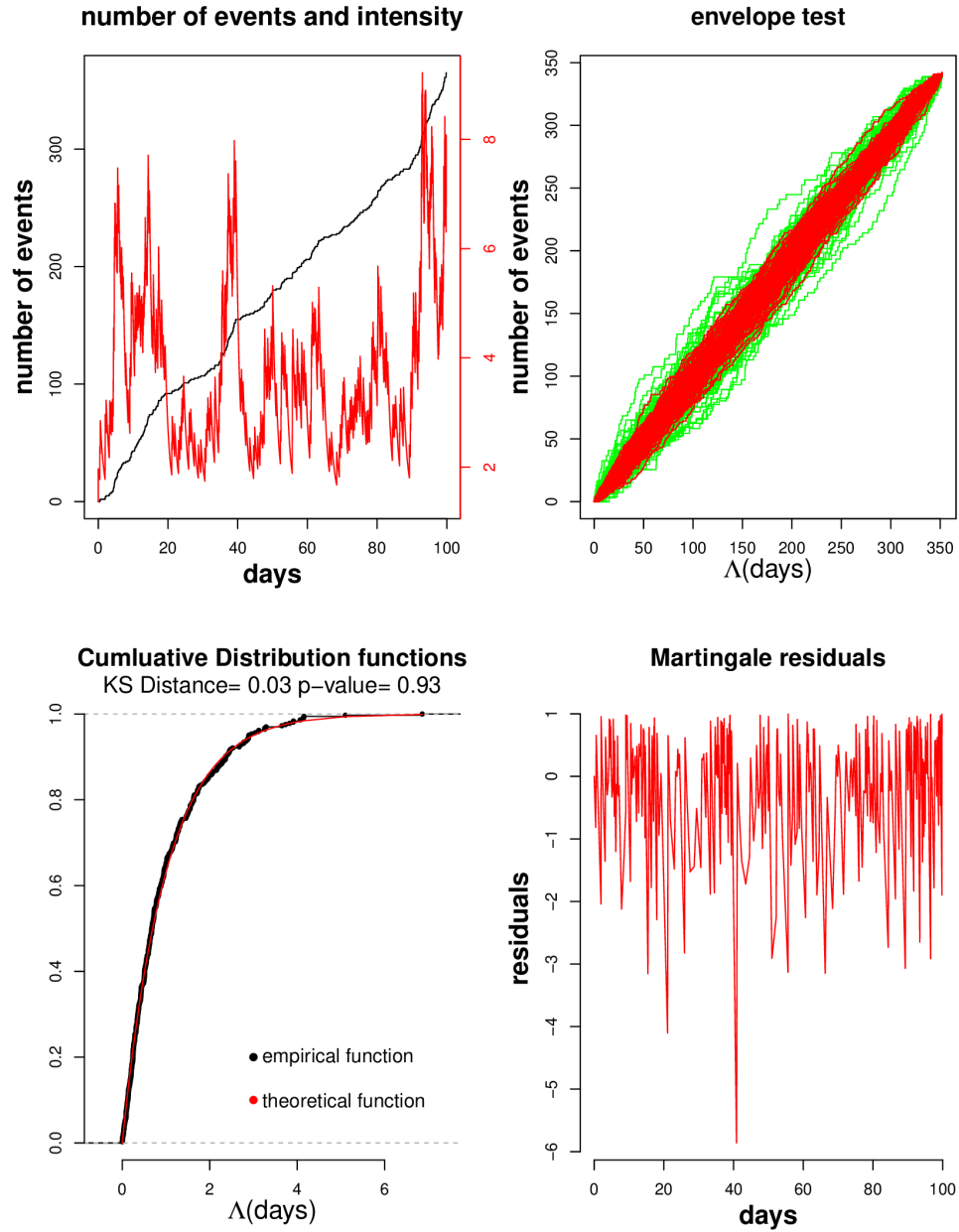


FIGURE 15 – Procédures d'inférence sur les données simulées : Courbe d'intensité conditionnelle superposée à la courbe d'événements, test d'enveloppe (les courbes vertes correspondent aux processus de Poisson à taux unitaire et les courbes rouges aux résidus des processus de Hawkes simulés), test de Kolmogorov-Smirnov et résidus de martingale.

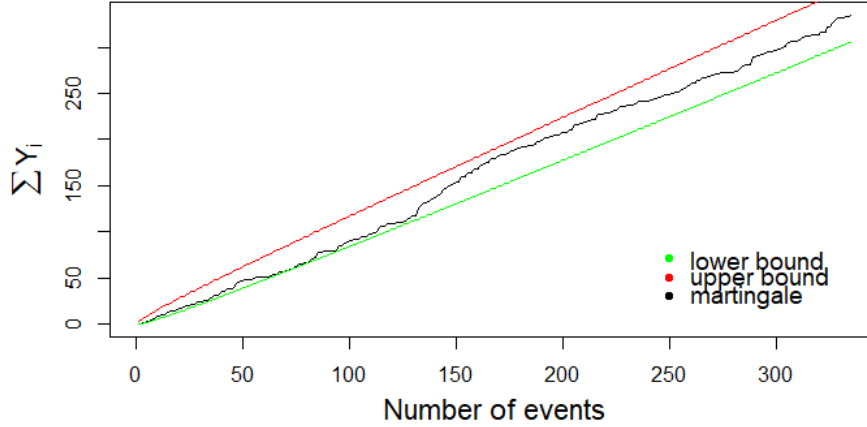


FIGURE 16 – Martingale des résidus et sa bande de confiance pour des données simulées selon le modèle exponentiel.

10.2 Application à des données sismiques réelles

10.2.1 Présentation des données

On travaille sur des données sismiques de la Guadeloupe sur les années 2004 - 2005 obtenues par l'équipe RING de GeoRessources. Le 21 Novembre 2004, a eu lieu le séisme le plus fort jamais enregistré instrumentalement à la Guadeloupe. Sa magnitude était de 6,3. La forte énergie qu'il a relâché dans le sol a provoqué d'autres répliques sismiques. Cette séquence sismique composée du séisme principal et des répliques peut être modélisée par un processus de Hawkes.

Nous disposons des temps d'occurrence des séismes (en secondes), sur l'intervalle de temps compris entre le 01/01/2004 et le 01/01/2006, ainsi que les magnitudes des séismes et les localisations des épicentres. L'instant à partir duquel les événements commencent à se multiplier correspond à la date du séisme du 21 Novembre 2004.

Les figures 17 et 18 montrent respectivement la distribution spatiale des épicentres des séismes en prenant tous les séismes et en ne considérant que ceux ayant une magnitude supérieure ou égale à $M_0 = 2$. Le choix d'éliminer les séismes de magnitude inférieure à 2 est lié au fait que ces séismes ont moins de chances de générer des descendants, on reviendra plus en détail sur ce point dans la section 12.4 sur le modèle ETAS. On remarque que cette distribution n'est pas homogène et que les séismes sont concentrés dans une région au milieu.

La figure 19 montre les valeurs des magnitudes en fonction des dates des séismes. On voit qu'il y a eu d'autres séismes de magnitude très importante

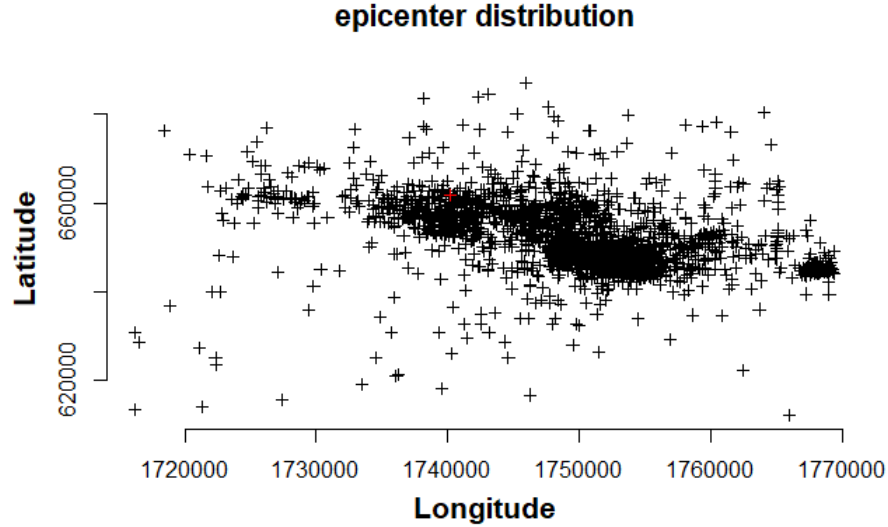


FIGURE 17 – Distribution des épicentres des séismes. Le point rouge correspond au séisme du 21/11

notamment autour des jours 80 et 200.

La figure 20 donne la fonction de répartition empirique des magnitudes en ne prenant en compte que les séismes ayant une magnitude supérieure à 2. L'allure de cette distribution ressemble à celle d'une loi exponentielle. On verra dans la section 12 de la deuxième partie sur les processus marqués qu'on peut considérer que la distribution des magnitudes suit une loi exponentielle décalée par une valeur seuil M_0 d'un paramètre β à estimer. On a tracé dans la figure 21 la distribution des magnitudes décalées superposée à la fonction de répartition exponentielle afin de comparer. Ces fonctions ont une allure similaire mais elles sont un peu éloignées ce qui revient au fait que les magnitudes peuvent être non indépendantes ou même dépendre du temps.

10.2.2 Estimation des paramètres

On applique une estimation par maximum de vraisemblance à l'échantillon à partir du 21/11/2004. On ramène ces dates au temps 0, en considérant le séisme du 21 novembre comme le premier événement arrivé. On transforme aussi l'échelle de secondes à des jours pour garantir la stabilité numérique de l'optimisation, ce qui correspond à 2229 dates sur une période $T = 405.23$ jours.

L'estimation nous donne :

$$\hat{\lambda} = 1.76, \hat{\alpha} = 3.44, \hat{\beta} = 5.06.$$

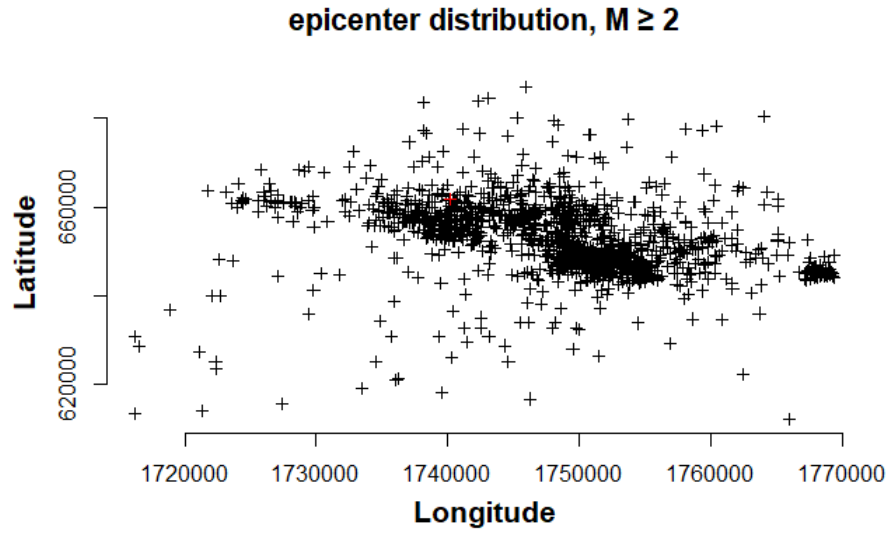


FIGURE 18 – Distribution des épicentres des séismes ayant une magnitude supérieure à 2. Le point rouge correspond au séisme du 21/11

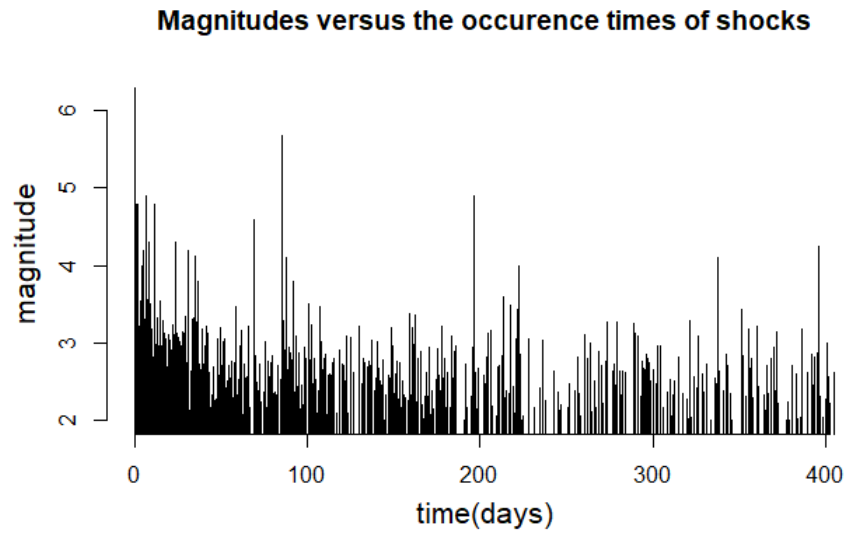


FIGURE 19 – Magnitudes en fonction des dates des séismes

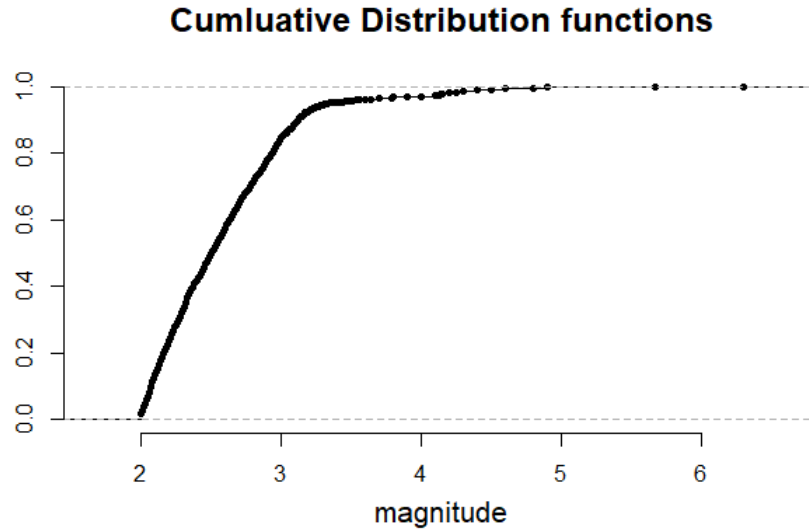


FIGURE 20 – Fonction de répartition empirique des magnitudes des séismes, la magnitude seuil est $M_0 = 2$.

Le code ci-dessous donne le code R pour réaliser cette estimation à partir des données.

```

donnees<-read.table("Donnees.txt",header=T)
dates=rev(donnees[,2]) #les donnees sont de base tries par ordre
                        dcroissant
m=length(dates)
x=donnees[donnees$m==6.3,2] #la date du seisme du 21/11/2004
i=which(dates==x) #i=37 indice de x dans dates
t2=dates[(i-1):m] #vecteur des dates partir du sisme du 21/11

#Donnes ramenes au temps 0, transforms en heure
t3=(t2-t2[1])/(3600*24) #transformation en chelle journalire

###Estimation des parametres###

est=estimation(t3)

#parametres estimes
lambdaes=est[1]; alphaes=est[2]; betaes=est[3]

```

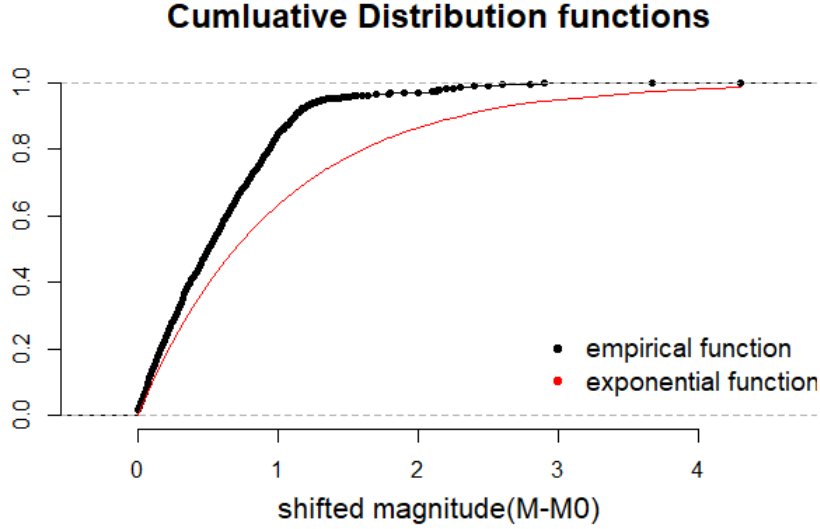


FIGURE 21 – Fonction de répartition empirique et théorique des magnitudes décalées des séismes.

10.2.3 Validation du modèle

Pour vérifier la validité du modèle, on effectue une analyse résiduelle. La figure 22 donne une présentation synthétique des procédures d'inférence mises en oeuvre.

Les résidus du test de l'enveloppe sont tous à l'intérieur de l'enveloppe, ce qui confirme qu'il s'agit de processus de Poisson à taux unitaire. Les résidus de martingale oscillent autour de 0 et s'éloignent légèrement de 0 vers le jour 110. Concernant la courbe du test de Kolmogorov-Smirnov, les deux fonctions de répartition, empirique et théorique, sont très proches, et la distance de Kolmogorov-Smirnov est égale à 0.03, qui est une petite valeur. Cependant, la p -valeur du test est égale à 0.02. Les résidus de martingale oscillent autour de 0 mais s'en éloignent à partir d'un certain rang. La figure 23 montre le tracé de la martingale $\sum Y_i$ de l'équation 19 et de sa bande de confiance. On voit qu'elle est au dessous de la borne supérieure de la bande mais qu'elle sort de la borne inférieure. Ces résultats nous encouragent à poursuivre le travail de modélisation basé sur les processus ponctuels de Hawkes en utilisant également les informations fournies par les magnitudes.

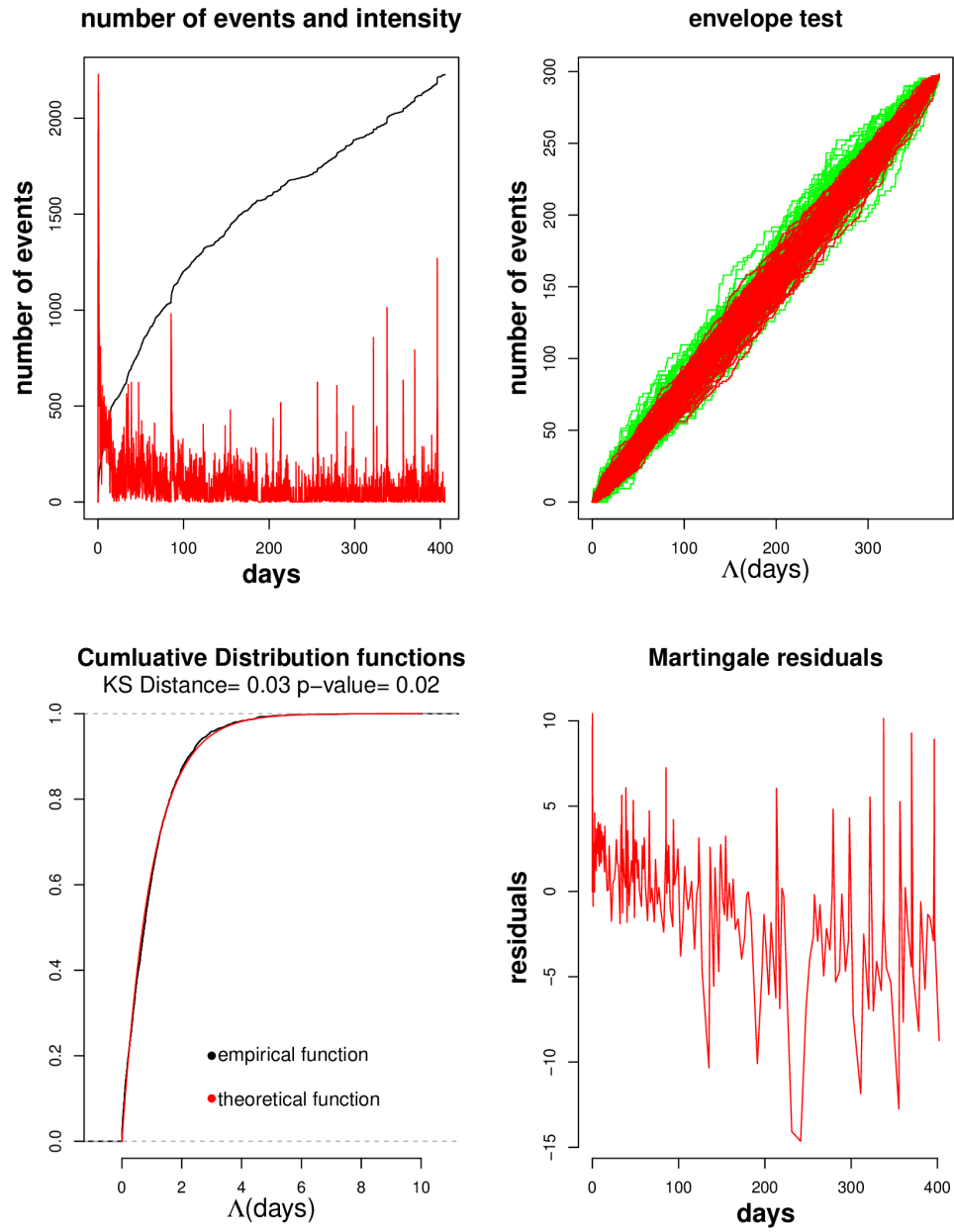


FIGURE 22 – Procédures d'inférence sur les données sismiques : Intensité conditionnelle, test d'enveloppe, test de Kolmogorov-Smirnov et résidus de martingale

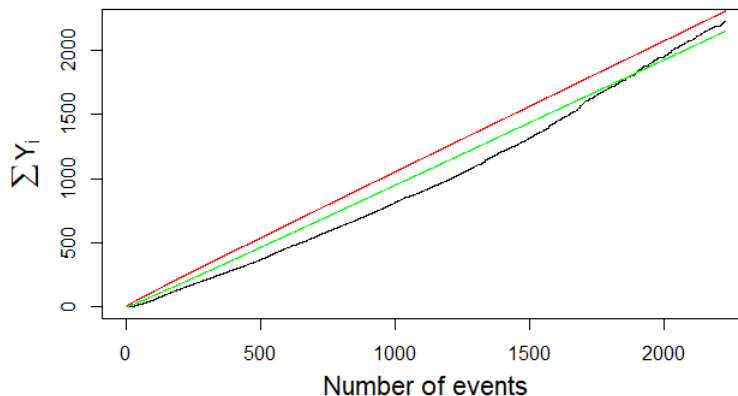


FIGURE 23 – Martingale des résidus et sa bande de confiance pour es données sismiques selon le modèle exponentiel.

Deuxième partie

Stage à l'IECL : Application à des modèles sismiques

Cette deuxième partie est constituée du travail que j'ai effectué pendant mon stage d'été à l'IECL. Il s'agit d'une extension des processus de Hawkes vus dans la première partie, en utilisant le modèle puissance et puis en passant aux cas marqués et spatio-temporels. Ces modèles sont motivés par des applications aux données sismiques.

11 Modèle de la fonction puissance

Dans cette section, on va étudier le modèle des processus de Hawkes avec la fonction puissance comme fonction d'excitation à la place d'une exponentielle. L'intensité due à un événement décroît alors moins rapidement.

11.1 Modélisation

Définition 7. Le modèle puissance des processus de Hawkes, correspond aux processus de Hawkes avec la fonction d'excitation :

$$\mu(t) = \frac{K}{(c+t)^p}, \quad (20)$$

avec $K, p \geq 0$ et c un réel strictement positif. Dans ce cas, l'intensité du processus s'écrit :

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} \frac{K}{(c + t - t_k)^p}.$$

Le paramètre p caractérise la décroissance de la fonction d'excitation, c est un décalage dans le temps et K est lié à l'amplitude de la contribution des événements.

Proposition 6. *Pour que le modèle soit bien défini, il faut que*

$$p > 1 \text{ et } (p-1)c^{p-1} > K. \quad (21)$$

Démonstration. D'après la proposition 5, pour que le modèle soit bien défini, il faut que :

$$\int_0^\infty \mu(s) ds < 1,$$

d'abord, pour que l'intégrale soit bien définie, il faut que $p > 1$. Et on a :

$$\int_0^\infty \mu(s) ds = \int_0^\infty \frac{K}{(c+s)^p} ds = \frac{K}{p-1} \frac{1}{c^{p-1}}.$$

D'où le résultat. □

11.2 Simulation

On simule les processus de Hawkes du modèle puissance en utilisant l'algorithme d'Ogata de la section 7.2 en utilisant la fonction d'excitation (20). Le code ci-dessous donne le programme R pour faire cette simulation.

```
sim.Hawkes.puissance=function(k,c,p,lambda,T){
  t=c();s=0      #t est le vecteur des temps d'occurrence du processus
  intensity.local=function(u,t,lambda){
    # fonction d'intensité d'un processus de Hawkes locale aux algorithmes
    # de simulation
    #t vecteur des dates des vnements arrivs jusqu' l'instant u
    n=0
    for (d in t){
      n=n+k/(c+u-d)^p
    }
    return(lambda+n)
  }
  lambda.sup=intensity.local(s,t,lambda)
  while (s<T){
    w=rexp(1,lambda.sup) #l'inter-arrivee
    s=s+w                #nouveau point candidat
    u=runif(1,0,1)
  }
```

```

lambda.temp=intensity.local(s,t,lambda)
if (u<lambda.temp/lambda.sup){ #on accepte s avec la proba
  lambda.temp/lambda.sup
  t=c(t,s) #on actualise le vecteur des dates d'occurence
}
lambda.sup=lambda.temp #on modifie lambda.sup chaque itration
}
if (t[length(t)]<=T) {
  N=c(0:length(t))
  t=c(0,t)
}
else {
  p=length(t)-1
  N=c(0:p)
  t=c(0,t[1:p])
}
list (time=t,nbpoints=N)
}
#commandes d'execution pour un exemple
set.seed(2020)

k=0.54;c=1.5;p=2;lambda=1;T=100;
res=sim.Hawkes.puissance(k,c,p,lambda,T)
plot(res$time,res$nbpoints,type="s",ylab="N(t)",col="blue",
      main="Realization of Hawkes point process – power law excitation
          function")

```

La figure 24 montre la réalisation d'un processus de Hawkes suivant le modèle puissance.

11.3 Inférence statistique

Proposition 7. *Pour une suite de dates d'occurrences $(t_i)_{1 \leq i \leq k}$, la log-vraisemblance d'un processus de Hawkes suivant le modèle puissance s'écrit sur $[0, t_k]$:*

$$\ell = \sum_{i=1}^k \ln(\lambda + K \sum_{j=1}^{i-1} \frac{k}{(c + t_i - t_j)^p}) - \lambda t_k - \frac{K}{1-p} \sum_{i=1}^{k-1} \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right].$$

Démonstration. D'après l'équation (11.3), la log-vraisemblance du processus s'écrit :

$$\ell = \sum_{i=1}^k \log(\lambda^*(t_i)) - \int_0^T \lambda^*(u) du.$$

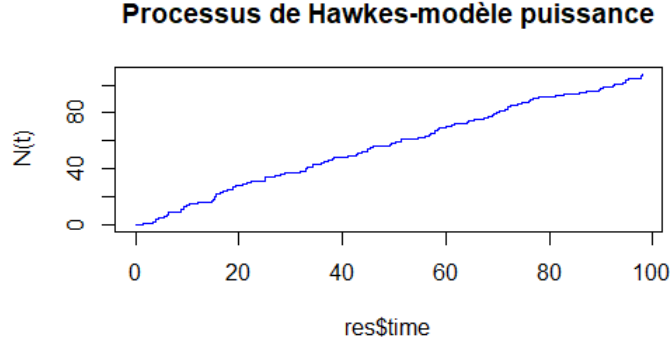


FIGURE 24 – Processus de Hawkes suivant le modèle puissance de paramètres $k = 0.54, c = 1.5, p = 2$, et $\lambda = 1$

On peut simplifier l'expression de cette fonction sur $[0, t_k]$:

$$\begin{aligned}
\int_0^{t_k} \lambda^*(u) du &= \int_0^{t_1} \lambda^*(u) du + \sum_{i=1}^{k-1} \int_{t_i}^{t_{i+1}} \lambda^*(u) du \\
&= \int_0^{t_1} \lambda du + \sum_{i=1}^{k-1} \left[\int_{t_i}^{t_{i+1}} \lambda + \sum_{t_j < u} \frac{K}{(c + u - t_j)^p} du \right] \\
&= \lambda t_k + K \sum_{i=1}^{k-1} \int_{t_i}^{t_{i+1}} \sum_{j=1}^i \frac{1}{(c + u - t_j)^p} du \\
&= \lambda t_k + K \sum_{i=1}^{k-1} \sum_{j=1}^i \int_{t_i}^{t_{i+1}} \frac{1}{(c + u - t_j)^p} du \\
&= \lambda t_k + \frac{K}{1-p} \sum_{i=1}^{k-1} \sum_{j=1}^i \left[\frac{1}{(c + t_{i+1} - t_j)^{p-1}} - \frac{1}{(c + t_i - t_j)^{p-1}} \right] \\
&= \lambda t_k + \frac{K}{1-p} \sum_{i=1}^{k-1} \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{(c + t_i - t_i)^{p-1}} \right] \\
&= \lambda t_k + \frac{K}{1-p} \sum_{i=1}^{k-1} \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right].
\end{aligned}$$

Donc

$$\ell = \sum_{i=1}^k \ln(\lambda + K \sum_{j=1}^{i-1} \frac{k}{(c + t_i - t_j)^p}) - \lambda t_k - \frac{K}{1-p} \sum_{i=1}^{k-1} \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right].$$

□

Le code ci-dessous donne le programme R pour calculer cette vraisemblance et pour la tracer en fonction de deux paramètres. On remarque que cette fonction est de complexité quadratique, contrairement au cas de la fonction d'excitation exponentielle, où on a pu simplifier l'expression pour se ramener à une complexité linéaire. Ce code renvoie l'opposé de la log-vraisemblance, car on cherchera à faire une minimisation dans la suite.

```
log_likelihood_puissance=function(x,t){ #t: le vecteur des temps
  d'occurrence,x=c(K,c,p,lambd)
  K=x[1];c=x[2];p=x[3];lambda=x[4]
  if (p<1 | (p-1)*c^(p-1)<K){return(0)}
  n=length(t)
  a1=rep(0,n)
  for (i in 2:n){
    t_inf=t[t<t[i]]
    a1[i]=sum(1/(c+t[i]-t_inf)^p) #A(i)
  }
  l= -t[n]*lambda
  -sum(1/(c+t[n]-t)^(p-1)-1/c^(p-1))*K/(1-p)+sum(log(lambda+K*a1))
  #oppos de la log-vraisemblance
  return(-l)}
#Pour tracer la log-vraisemblance:
x <- seq(0.1,4,by=0.2) #k
y <- seq(0.1,4,by=0.2) #lambda
#Les vrais parametres de l'echantillon
k=0.5;c=1.5;p=1.5;lambda=2;T=100
t=sim.Hawkes.puissance(k,c,p,lambda,T)$time
z=matrix(rep(0, length(x)*length(y)), length(x), length(y))
for(i in 1:length(x)){
  for (j in 1:length(y)){
    z[i,j]=-log_likelihood_puissance(c(x[i],c,p,y[j]),t)
  }
}

pmat<-persp(x,y,z, xlim = range(x), ylim = range(y),zlim = range(z, na.rm
= TRUE),xlab="k",ylab="lambda", zlab ="lnL",theta = 20, phi =
20,scale=TRUE, axes=T, ticktype="detailed")

#On ajoute au graphe le point qui represente les vrais paramtres de
l'echantillon
e=c(k,c,p,lambda)
x0=k;y0=lambda;z0=-log_likelihood_puissance(e,t)
trans3d <- function(x,y,z, pmat) {
  tr <- cbind(x,y,z,1) %*% pmat
  list (x = tr[,1] /tr[,4], y= tr[,2] /tr[,4])
}
```



```
}
points(trans3d(x0,y0,z0,pmat),pch=16, col="red")
```

Les figures 25 et 26 donnent respectivement le tracé de la log-vraisemblance en fonction de k et λ , et puis en fonction de c et p . On observe que la log-vraisemblance est concave dans le premier cas, mais elle a tendance à s'aplatir dans le deuxième cas. **Estimation des paramètres** On cherche à estimer les

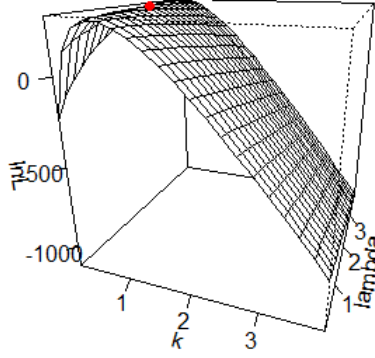


FIGURE 25 – La log-vraisemblance d'un échantillon de Hawkes de paramètres $k = 0.5, c = 1.5, p = 1.5, \lambda = 2$ avec $T = 100$ en fonction des paramètres k et λ . Le point en rouge représente les vrais paramètres.

paramètres K, c, p et λ , il s'agit d'un problème d'optimisation avec des contraintes non linéaires d'après l'équation (21). On peut l'écrire sous la forme suivante :

$$\begin{cases} \min_{K, c, p, \lambda} \ell \\ \text{s.c } \lambda \geq 0, K \geq 0, p > 1, c > 0, \\ (p-1)c^{p-1} > K. \end{cases} \quad (22)$$

Daley et Vere-Jones [3] ont montré dans l'exemple 7.2(c) que la log-vraisemblance est concave en fonction des paramètres pour lesquels la dépendance de l'intensité conditionnelle est linéaire. Dans notre cas, elle sera concave en fonction de K et c . Donc on peut envisager de faire une estimation par fixation de paramètres qui consiste à fixer c et p et à estimer K et λ en maximisant la log-vraisemblance selon ces deux paramètres, puis on choisit le meilleur mo-

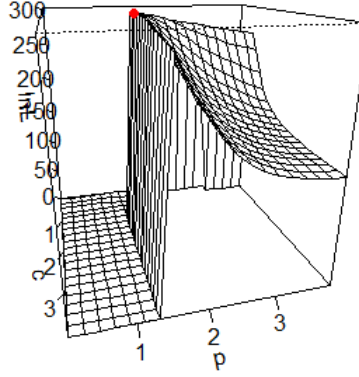


FIGURE 26 – La log-vraisemblance d’un échantillon de Hawkes de paramètres $k = 0.5, c = 1.5, p = 1.5, \lambda = 2$ avec $T = 100$ en fonction des paramètres c et p . Le point en rouge représente les vrais paramètres.

dèle par le critère d’Akaike. L’optimisation sera stable dans ce cas vu que la log-vraisemblance sera concave.

Dans notre cas, on fait varier p entre 1.25 et 2.25 avec un pas égal à 0.25, et on fait varier c entre 0.5 et 2.5 avec un pas égal à 0.5. Donc on aura 25 modèles au final.

Si on choisit les valeurs de p plus grandes on risque de trop pénaliser la fonction d’excitation. Le choix des valeurs de c est lié au fait qu’il s’agit d’un décalage dans le temps qui garantit la contribution des événements proches et que si on s’intéresse à des données journalières un décalage entre 0.5 jours et 2.5 jours semble raisonnable.

Les étapes de l’optimisation sont les suivantes :

- on crée une liste de valeurs de c et p dans les intervalles précisés ci-dessus.
- Pour chaque valeur de c et p
 - On maximise la log-vraisemblance par rapport à λ et k et on calcule la valeur du AIC.
- on retient le modèle (les valeurs de c, p, λ et k) qui a le plus petit AIC.

Le programme ci-dessous donne le code R pour faire une estimation par maximum de vraisemblance par fixation des paramètres c et p

```

#Fonctions pour trouver des parametres qui verifient la contrainte du
  modele puissance
find_c=function(k,p){
  u=p-1
  return((k*1.0001/u)^(1/u))
}
find_k=function(c,p){
  u=(p-1)*c^(p-1)
  k=u-0.001
  return(k)
}

#code pour l'estimation de k et lambda pour des parametres c et p fixes

estimation_puissance3=function(c,p,temps){ #temps est le vecteur des dates
  d'occurrence du processus qui nous interesse
  #objective function=log-vraisemblance comme fonction d'un seul parametre
  lnL=function(x){
    #x:k,c,p,lambda
    k=x[1];lambda=x[2]
    return(log_likelihood_puissance(c(k,c,p,lambda),temps))
  }
  #il faut  $k < (p-1)*c^{p-1}$  donc il faut  $-k \geq 0.01 - (p-1)*c^{p-1}$ 
  const=(p-1)*c^(p-1)-0.0001
  k0=find_k(c,p) #valeur initiale de k qui respecte la contrainte
  v=constrOptim(theta=c(k0,1),f=lnL,grad=NULL,ui=rbind(c(1,0),c(-1,0),c(0,1)),ci=c(0,-const,0))
  return(c(v$par,v$value)) #le code renvoie l'optimum et la valeur de logL
  cet optimum
}

#Application de l'estimation par fixation de parametres un echantillon
  simule

#echantillon simule
k=0.6;c=1.5;p=1.5;lambda=2;T=100
proc=sim.Hawkes.puissance(k,c,p,lambda,T)
t=proc$time
#on aura besoin de t pour l'analyse des residus
#on peut l'enregistrer dans un fichier csv
#write.csv(t, file = "vecteur_t.csv")
N=proc$nbpoints
#test:
estimation_puissance3(c,p,t)

```

```

#liste des valeurs de c et p
val _p=seq(1.25,2.25,by=0.25) #5 valeurs
val _c=seq(0.5,2.5,by=0.5)
#on cherche minimiser -logL et on choisit le modele avec AIC le plus
    petit
#ici on prend la valeur de logL pour AIC: tous les modeles ont le meme
    nombre de parametres
#xoptim contient les valeurs estimees du meilleur modele
minAIC=10^5;xoptim=c(0,0,0,0)

#tableau pour enregistrer les valeurs estimees :
L2=data.frame(init=c(0,0,0,0))
rownames(L2)=c("p","c","k","lambda","AIC")

#pour chaque c et p on estime et ajoute les valeurs L2
# a peut prendre un peu de temps ~3min
for (p in val _p){
  for (c in val _c){
    v=estimation_puissance3(c,p,t) #de taille 3: k, lambda et -logL
    x=v[1:2];obj=v[3] #x contient k et lambda, obj=-logL
    AIC=obj
    v2=c(p,c,x[1],x[2]) #vecteur de tous les parametres
    L2=cbind(L2,valeur=c(v2,AIC))
    if (AIC<minAIC){ #on modifie minAIC et xoptim si on trouve un
      meilleur modele
      minAIC=AIC
      xoptim=v2
    }
  }
}
#l'estimation selon le meilleur modele est :
xoptim #p,c,k,lambda

#en visualisant L2
data=L2
sorted2=data[,order(data["AIC",])]
#on utilisera sorted2 dans l'analyse residuelle
tsort=t(sorted2)
#en visualisant tsort on verra les modeles tries par ordre decroissant du
    AIC

```

11.4 Validation du modèle

Pour valider le modèle, on effectue l'analyse résiduelle du théorème 7. En suivant la même démarche qu'on a suivi pour trouver l'équation (18), on trouve

que l'intensité cumulée du modèle puissance s'écrit :

$$\Lambda(t) = \int_0^t \lambda^*(s) ds = \lambda t + \frac{K}{1-p} \sum_{t_i < t} \left[\frac{1}{(c+t-t_i)^{p-1}} - \frac{1}{c^{p-1}} \right]$$

Les codes suivants donnent les programmes R pour réaliser : une analyse des résidus avec le test d'enveloppe sur des échantillons de Hawkes simulés avec le modèle de la fonction puissance, un test de Kolmogorov-Smirnov et un test de martingale de résidus.

```

#l'intensite cumulee d'un processus de Hawkes temps calculee au point s
int_cumul_puissance=function(temps,s,lambda,k,c,p){
  A=temps[temps<s]
  return(lambda*s+(k/(1-p))*sum(1/((c+s-A)^(p-1))-(1/c^(p-1)))) }

#Fonction pour calculer les residus
residu_puissance=function(t,lambda,k,c,p){
  res=rep(0,length(t))
  for (i in 2:length(res)){
    temp=t[i]
    res[i]=int_cumul_puissance(t,temp,lambda,k,c,p)
  }
  return(res)}

#test d'enveloppe
envelope_residuals_puissance=function(T,n,lambda,k,c,p){
  process=sim.Hawkes.puissance(k,c,p,lambda,T)
  v=process$time
  res=residu_puissance(v,lambda,k,c,p)
  N=process$nbpoints
  plot(res,N,type="s",col="green",ylab="N(t)")
  for (i in 1:n){ #on trace les processus de Poisson(1)
    par(new = T)
    v=Poisson_homogene(T,1)
    N=c(0:(length(v)-1))
    plot(v,N,type="s",col="green",axes=F,xlab="",ylab="")
  }
  for (i in 1:n){ #on trace les residus des processs de Hawkes simules
    par(new = T)
    process=sim.Hawkes.puissance(k,c,p,lambda,T)
    v=process$time
    res=residu_puissance(v,lambda,k,c,p)
    N=process$nbpoints
    plot(res,N,type="s",col="red",axes=F,xlab="",ylab="")
  }
}

```

```

#Commandes d'excution
k=1.2;c=20;p=1.5;lambda=2;T=100
t=sim.Hawkes.puissance(k,c,p,lambda,T)$time
envelope _residuals _puissance(80,100,lambda,k,c,p)



---



#Test de kolmogrov Sminrov
#Kolmogorov-Sminrov
res=residu _puissance(t,lambdaes,k,c,p)
diff _sim=c(res[1],diff(res))
p_val=ks.test(diff _sim,"pexp",1)
p_val
D=p_val$statistic

#Graphe des fonctions de rpartition empirique et thorique
inter=sort(diff _sim)
plot(ecdf(inter),ylab="",xlab=expression(paste(Lambda,"(days)")),main =
'Cumulative Distribution functions',cex.main=0.9,pch=20)
legend("bottomright",legend=c("empirical function", "theoretical
function"),col=c("black", "red"), lty=1,1, cex=0.65)
mtext(paste("KS Distance=", round(D,2)),side=3,cex=0.75)
#par(new=T)
lines(inter , pexp(inter,1),ylab="",col="red",type="l")



---



#Test de rsidus de martingale

residus _mar _puissance=function(ind,t,lambda,k,c,p){
  resid=rep(0,length(ind))
  for (i in 2:length(resid)){
    #il y a ind[i] vnements arrivs jusqu' t[ind[i]]
    resid[i]=ind[i]-int _cumul _puissance(t,t[ind[i]],lambda,k,c,p) # rsidu
    l'instant t[ind[i]]
  }
  return(resid)
}
#creons une grille z partir de t
n=200
ind=seq(1,length(t),by=floor(length(t)/n))
z=rep(0,length(ind))
for (i in 1:length(z)){
  z[i]=t[ind[i]]
}
#commande d'excution
res _z=residus _mar _puissance(ind,t,lambda,k,c,p)
#liste des differences

```

```
diff_z=c(res_z[1],diff(res_z))
#####GRAPHE4#####
plot(z, diff_z,xlab="days",ylab="residuals",main = 'Martingale
residuals—power law',cex.main=0.9,type='l',col=2)
```

11.5 Application à des données simulées

On simule un échantillon de Hawkes de paramètres $k = 0.54, c = 1.5, p = 1.5$ et $\lambda = 2$ par l'algorithme d'Ogata. On effectue une estimation par fixation des paramètres comme expliqué dans la section précédente. On fixe les paramètres p et c et on maximise numériquement la log-vraisemblance en fonction de k et λ , qui sera concave en fonction de ces paramètres. On fait varier p entre 1.25 et 2.5, pour respecter la contrainte sur p et vu qu'une valeur plus grande de p risque de trop pénaliser l'effet des événements. c correspond à un décalage dans le temps qui garantit la contribution des événements proches à l'intensité, on s'intéresse à des données journalières donc on fait varier c entre 0.5 et 2.5 jours. Au final on obtient 25 modèles.

L'estimation par fixation de paramètres a renvoyé les valeurs suivantes, qui correspondent au meilleur modèle parmi les 25 par critère d'Akaike :

$$\hat{k} = 0.57, \hat{c} = 1.5, \hat{p} = 1.5, \hat{\lambda} = 1.98.$$

Les valeurs estimées sont proches des vrais paramètres. La figure 29 montre le résultat des estimations des 5 meilleurs modèles selon le critère d'Akaike. Le premier modèle renvoie les valeurs les plus proches des vrais paramètres, ainsi on a effectué une analyse résiduelle sur ce modèle. La figure 28 montre le résultat de cette analyse. On observe que le test d'enveloppe valide le modèle

	p	c	k	lambda	AIC
valeur	1.50	1.5	0.5681426	1.984851	-624.9868
valeur.1	1.75	1.5	0.8322837	1.930160	-624.9839
valeur.2	1.75	1.0	0.5833176	2.014061	-624.7475
valeur.3	1.75	2.0	1.0623625	1.936239	-624.7418
valeur.4	2.00	2.0	1.5793142	1.910019	-624.7068

FIGURE 27 – Les 5 meilleurs modèles d'estimation des paramètres d'un échantillon de Hawkes selon le modèle puissance par la méthode de fixation des paramètres. Les vrais paramètres sont $k = 0.54, c = 1.5, p = 1.5, \lambda = 2$.

puisque les résidus tracés en rouge sont bien dans l'enveloppe de Poisson. Le test de Kolmogorov valide aussi le modèle, sa p-valeur est de l'ordre de 0.94 et les fonctions de répartition théorique et empirique sont assez proches. Le test des résidus de martingales valide aussi le modèle, car ces résidus oscillent autour de 0, en plus on a montré dans la remarque 9 que l'intervalle de confiance de la

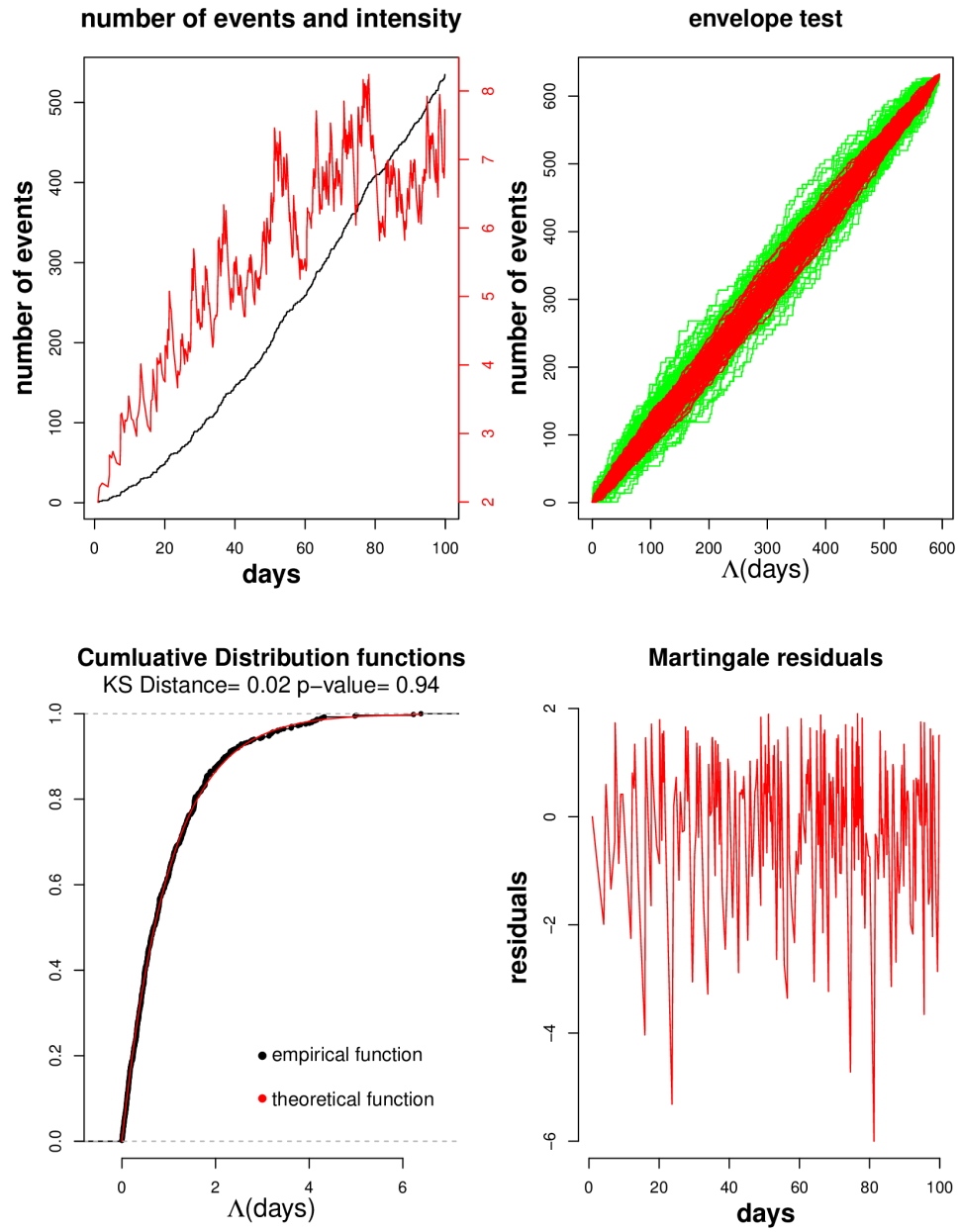


FIGURE 28 – Analyse résiduelle appliquée à un échantillon de Hawkes simulé selon le modèle puissance.

somme des résidus au risque 5% est $[-15.86, 16.99]$, donc on est bien dans cet intervalle de confiance.

11.6 Application à des données réelles

On applique le modèle puissance aux données sismiques de la Guadeloupe. L'estimation par fixation des paramètres selon le meilleur modèle donne :

$$\hat{k} = 1.15, \hat{c} = 1, \hat{p} = 2.25, \hat{\lambda} = 0.48.$$

On peut interpréter la valeur de λ en disant qu'en moyenne, il y a 0.48 nouveaux séismes qui arrivent, on remarque que cette valeur est inférieure à celle qu'on a trouvé pour le modèle exponentiel, à savoir 1.76.

	p	c	k	lambda	AIC
valeur	2.25	1.0	1.1475660	0.4761032	-2420.984
valeur.1	2.00	1.0	0.9594434	0.2774864	-2397.831
valeur.2	2.25	1.5	1.9733127	0.3075958	-2371.086
valeur.3	1.75	1.0	0.7498998	0.1116500	-2363.188
valeur.4	2.00	1.5	1.4780225	0.1575599	-2342.787

FIGURE 29 – Les 5 meilleurs modèles d'estimation des paramètres des données sismiques selon le modèle puissance par la méthode de fixation des paramètres.

La figure 30 donne le résultat de l'analyse résiduelle. Concernant le test d'enveloppe les résidus sont globalement dans l'enveloppe de Poisson mais sortent un peu de cette enveloppe pour certains échantillons. Pour le test de Kolmogorov-Smirnov les fonctions de répartition théorique et empirique sont assez proches mais il renvoie une p-valeur très petite de l'ordre de 10^{-7} ce qui ne valide pas le modèle. Le test des résidus de martingale quant à lui valide le modèle, car les résidus oscillent autour de 0, en plus on a montré dans la remarque 9 que l'intervalle de confiance de la somme des résidus au risque 5% est $[-33.45, 32.32]$, donc on est bien dans cet intervalle de confiance.

12 Processus de Hawkes marqués

Quand les événements du processus possèdent une caractéristique observable, on peut l'introduire dans le modèle comme variable supplémentaire qu'on appelle la marque du processus. Ainsi, on parle de processus marqués $\{(t_i, m_i)\}$ sur $[0, T) \times \mathcal{M}$ où \mathcal{M} est l'espace des marques.

12.1 Modélisation

On parle de processus de base qui correspond au processus des dates des événements sans les marques. On note l'intensité conditionnelle de ce processus

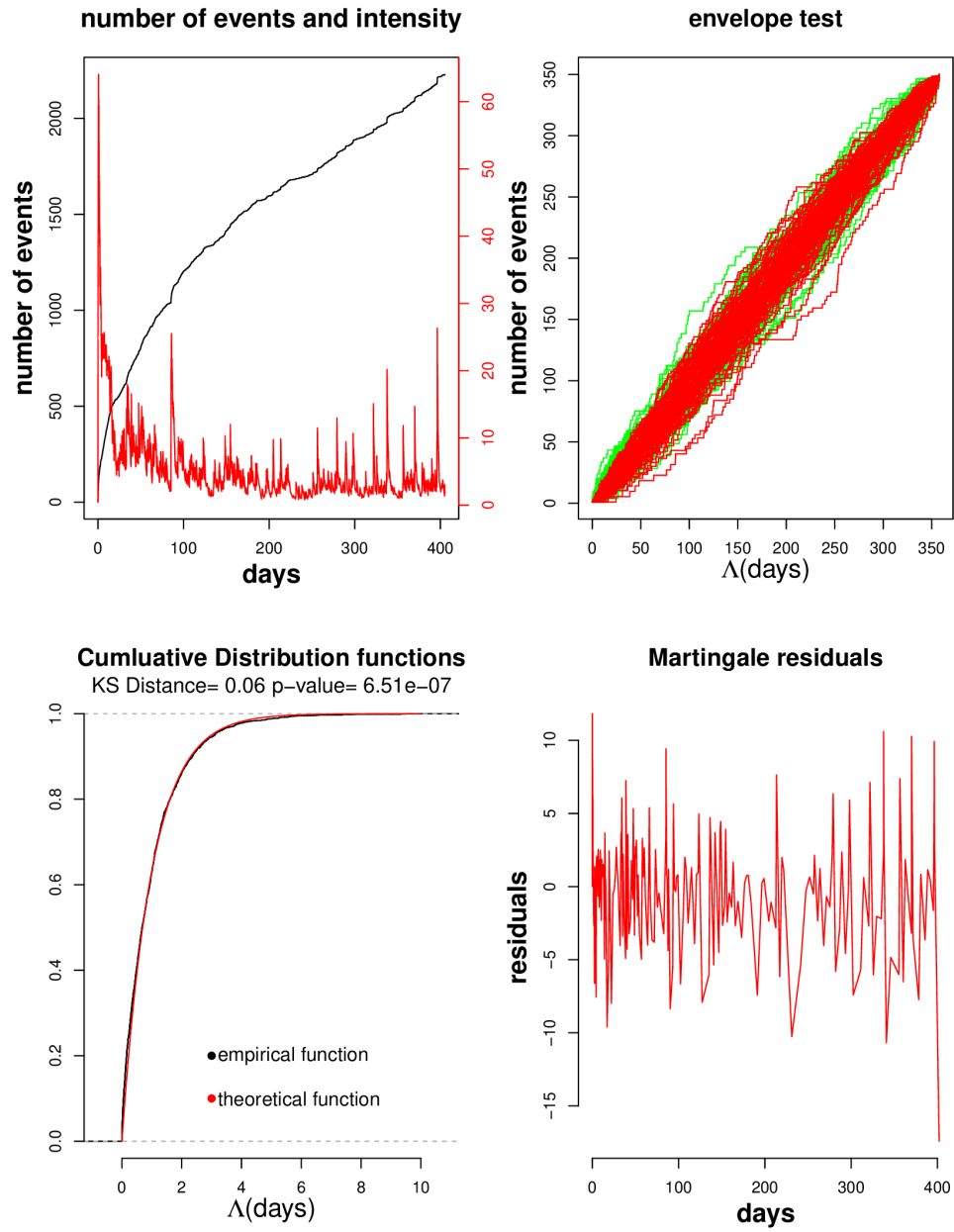


FIGURE 30 – Analyse résiduelle appliquée à aux données sismiques selon le modèle puissance.

$\lambda_g(t)$. Celle-ci peut dépendre du passé des marques ainsi que du passé des temps d'arrivée. Ainsi l'intensité conditionnelle du processus marqué est donnée par :

$$\lambda^*(t, m) = \lambda_g(t) \lambda(m|t)$$

où $\lambda(m|t)$ désigne l'intensité conditionnelle de la marque à l'instant t sachant le passé du processus jusqu'à cet instant. Ce passé est donné par $H_t = \{(t_i, m_i); t_i < t\}$.

On dit que les marques d'un processus sont indépendantes, si elles sont mutuellement indépendantes, et la distribution d'une marque m_i dépend uniquement de t_i . C'est le cas quand les m_i sont identiquement distribuées et indépendantes.

On dit que les marques sont imprévisibles, si m_i est indépendante de toutes les dates et les marques $\{t_j, m_j\}$ des événements précédents $t_j < t_i$. Ainsi le processus de base n'influence pas la distribution des marques, mais les marques peuvent influencer l'évolution du processus. Le cas le plus commun de processus avec des marques imprévisibles est celui où elles sont, conditionnellement au passé du processus, identiquement distribuées et indépendantes. Daley et Vere Jones (2003) [3] donnent dans les exemples (c) et (d) de la section 6.4 plus de détails sur les processus de Hawkes avec des marques imprévisibles.

Exemple : En sismologie, la magnitude des séismes peut être intégrée comme une marque. En effet, la distribution des répliques peut dépendre de ces magnitudes. Ainsi on observe des événements $\{(t_i, m_i)\}$ avec $m_i \geq M_0$ où M_0 est la magnitude seuil. L'intensité conditionnelle de ces processus marqués peut s'écrire, d'après Ogata [10], sous la forme :

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} c(m_k) \mu(t - t_k)$$

avec

$$c(m_k) = \exp(\alpha(m_k - M_0)).$$

Le coefficient b mesure l'effet de la magnitude sur la génération de nouvelles répliques, et μ est la fonction d'excitation définie auparavant. On peut prendre $\mu(t) = \alpha \exp(-\beta t)$ ou $\mu(t) = \frac{k}{(t+c)^p}$. On en reviendra plus en détail quand on discutera du modèle ETAS dans la section 12.4.

12.2 Simulation

Si les marques sont indépendantes du temps, et qu'on connaît leur densité, on peut les simuler séparément pour chaque événement, et on simule le processus de Hawkes par amincissement en utilisant l'expression de l'intensité conditionnelle.

Dans le cas où on s'intéresse à des données sismiques, avec des magnitudes indépendantes du temps et indépendantes entre elles. L'équation (24) de la section suivante, nous donne l'expression de la densité des magnitudes. On peut utiliser la méthode de l'inverse de la fonction de répartition pour simuler ces

magnitudes. Soit M_0 la magnitude seuil désirée, la densité est une fonction définie sur \mathbb{R}^+ qui dépend d'un paramètre β :

$$f(m) = \begin{cases} \beta \exp(-\beta(m - M_0)), & \text{si } m \geq M_0 \\ 0 & \text{sinon.} \end{cases}$$

Donc la fonction de répartition s'écrit

$$\begin{aligned} F(m) &= \int_0^m f(u) du = \int_{M_0}^m \beta \exp(-\beta(u - M_0)) du \\ &= 1 - \exp(-\beta(m - M_0)), \end{aligned}$$

l'inverse de cette fonction est donné par :

$$F^{-1}(m) = M_0 - \frac{\ln(1 - y)}{\beta}.$$

Ainsi si U est une variable uniforme sur $[0, 1]$ alors $F^{-1}(U)$ a une densité f .

On verra dans la suite que l'estimation de β est donnée par $\hat{\beta} = (\bar{m} - M_0)^{-1}$ tel que $\bar{m} = \frac{1}{n} \sum_{i=1}^n m_i$ et n est le nombre de séismes avec une magnitude supérieure à M_0 . Donc une valeur grande de β maintient les magnitudes autour de M_0 , alors qu'une valeur petite peut donner des valeurs éloignées de M_0 . On a simulé un processus de Hawkes marqué pour le cas du modèle ETAS dans la section 12.4.

12.3 Inférence

D'après Daley et Vere-Jones [3], la log-vraisemblance d'un processus de Hawkes marqué s'écrit :

$$L = \left[\prod_{i=1}^p \lambda^*(t_i, m_i) \right] \exp \left(- \int_0^T \int_{\mathcal{M}} \lambda^*(u, m) dm \right).$$

12.3.1 Marques indépendantes du temps

Dans le cas des marques indépendantes du temps, on peut décomposer la vraisemblance en le produit de deux termes indépendants :

$$L = \mathbb{P}(m_1, m_2, \dots | \eta) \mathbb{P}(t_1, t_2, \dots | H, \theta),$$

où η et θ sont respectivement les paramètres cachés de la distribution des marques et de la distribution des dates d'occurrence. Donc la maximisation de la vraisemblance totale revient à maximiser chaque terme.

- L'estimation de θ se fait en maximisant le deuxième terme de la vraisemblance dont la log-vraisemblance s'écrit, d'après l'équation (11.3) :

$$\ell = \sum_{i=1}^k \log(\lambda^*(t_i)) - \int_0^T \lambda^*(u) du. \quad (23)$$

En considérant l'intensité comme fonction uniquement du temps mais qui peut dépendre du passé des marques.

- L'estimation de η dépend des propriétés des marques. Nous présentons ci-dessous l'inférence quand on s'intéresse à des marques indépendantes en sismologie. D'après la formule de Gutenberg-Richter [5], l'intensité des magnitudes s'écrit :

$$\lambda(m) = 10^{a-bm} = A \exp(-\beta m),$$

avec, a et b des réels et $\beta = b \ln(10)$. En imposant une magnitude seuil M_0 , on peut en déduire la densité de probabilité des magnitudes

$$f(m) = \frac{\lambda(m)}{\int_{M_0}^{\infty} \lambda(m) dm} = \beta \exp(-\beta(m - M_0)). \quad (24)$$

La valeur de β est estimé par maximisation de la vraisemblance

$$L(\beta) = \prod_{i=1}^n f(m_i) = \beta^n \prod_{i=1}^n \exp(-\beta(m_i - M_0)),$$

avec n le nombre de séismes avec une magnitude supérieure à M_0 , ce qui donne

$$\hat{\beta} = (\bar{m} - M_0)^{-1},$$

avec $\bar{m} = \frac{1}{n} \sum_{i=1}^n m_i$.

Le programme ci-dessous donne le code R pour estimer β pour une suite de magnitudes connaissant la magnitude seuil M_0 .

```
estim_beta=function(magnitude,M0){
  n=length(magnitude)
  moy=sum(magnitude)*1/n
  return(1/(moy-M0))
}
```

On a appliqué ce résultat aux données sismiques de la Guadeloupe. La figure 31 montre le résultat du test QQplot qu'on a appliqué aux magnitudes réelles et à des magnitudes simulées de paramètres β donné par estimation sur les données sismiques pour $M_0 = 2$. Pour des valeurs petites, les quantiles théoriques sont égaux aux les quantiles empiriques, mais ces derniers s'éloignent quand la valeur de la magnitude augmente, ce qui signifie que la queue droite de la densité théorique est plus épaisse. Ceci peut s'expliquer par le fait que la vraie distribution des magnitudes ne suit pas exactement la forme exponentielle et tend vers zero plus rapidement quand la valeur de la magnitude augmente.

12.3.2 Marques qui dépendent du temps

On présente dans cette section l'approche d'inférence conduite par Ogata (1989) [9] pour le modèle de Hawkes avec des marques qui dépendent du temps.

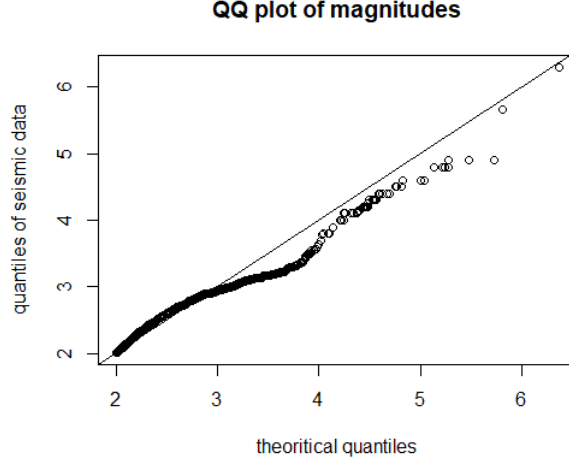


FIGURE 31 – QQpot des magnitudes des données sismiques et d'un échantillon simulé

Ogata a détecté quelques anomalies dans l'analyse résiduelle pour le modèle de marques indépendantes du temps, ainsi il a introduit un nouveau modèle pour vérifier si les magnitudes dépendent du passé des événements.

On applique la formule de Gutenberg-Richter avec un paramètre β qui dépend du temps. On introduit dans le modèle ETAS, qui sera définie dans la section suivante, la variable $\sigma(t) = \beta(t)^{-1}$, définie comme

$$\sigma(\tau) = \sigma(\tau|H(\tau)) = \mu + \sum_{\tau_i < \tau} g_J(\tau - \tau_i) + \sum_{\tau_j < \tau} h_K(\tau - \tau_j)m_j$$

où les τ sont les résidus définis dans le théorème 14. Les fonctions g_J et h_K sont paramétrées par les polynômes de Laguerre $g_J(t) = \sum_{j=1}^J a_j t^{j-1} \exp(-ct)$ et $h_K(t) = \sum_{k=1}^K b_k t^{k-1} \exp(-dt)$. La fonction de vraisemblance s'écrit :

$$L(\mu, a_1, \dots, a_J, b_1, \dots, b_K, d) = \prod_{i=1}^n \sigma(\tau_i)^{-1} \exp(-(m_i - M_0)/\sigma(\tau_i)).$$

On maximise cette fonction par rapport à $\theta = (\mu, a_1, \dots, a_J, b_1, \dots, b_K, d)$ en fixant J et K . Les ordres optimaux de J et K sont trouvés en minimisant le critère Akaike après estimation par maximum de vraisemblance.

Pour les données sur lesquelles a travaillé Ogata, il a trouvé que les valeurs optimales correspondent à $(J, K) = (1, 1)$ ce qui montre ce pour ces données, les magnitudes dépendent effectivement du passé des événements.

12.4 Modèle ETAS (Epidemic Type Afterschock Sequence model)

12.4.1 Définition et modélisation

Le modèle ETAS est bien connu sismologie statistique. C'est un cas des processus de Hawkes marqués[12]. Ce modèle découle de la loi modifiée d'Omori qui décrit la fréquence des répliques sismiques comme mentionné dans la remarque 2 :

$$\phi(t) = \frac{K}{(t+c)^p}, t > 0,$$

la valeur de p a été estimée pour plus de 200 séquences de répliques, elle est entre 0.6 et 2.5 avec une valeur médiane de 1.1.

En fait, cette expression ne décrit que la génération des descendants de la première génération, et ne prend pas en compte le cas où des séismes descendants forts engendrent à leur tour des répliques. Ogata a proposé de modéliser la densité de génération des descendants de chaque choc déclenchant i arrivé à l'instant t_i avec une magnitude m_i par :

$$\phi_i(t) = \frac{K_i}{(t-t_i+c)^p}, t > 0,$$

avec

$$K_i = K \exp(\alpha(m_i - M_0)), \quad (25)$$

les paramètres (K, α, c, p) sont communs pour toutes les répliques, M_0 est une magnitude de référence des données, inférieure ou égale à la magnitude seuil des séismes. Le poids de la fonction ϕ_i dépend de la magnitude du choc, ce qui montre que le nombre d'événements qu'il génère dépend de cette magnitude.

Définition 8. Un processus ETAS est un processus de Hawkes marqué, qui s'applique pour modéliser les répliques sismiques, ayant comme intensité conditionnelle :

$$\lambda^*(t) = \lambda + \sum_{t_i < t} K \frac{\exp(\alpha(m_i - M_0))}{(t-t_i+c)^p}, \quad (26)$$

où les t_i sont les dates d'occurrence des séismes et m_i sont leurs magnitudes. M_0 est une magnitude de référence inférieure aux m_i .

Proposition 8. Pour que le modèle ETAS soit bien défini il faut que

$$\alpha < \beta \text{ et } \frac{K}{(p-1)^{p-1}} \frac{\beta}{(\beta-\alpha)} < 1$$

Démonstration. Voir la proposition 6.4 de Daley et Vere-Jones [3]. \square

Ogata [12] s'est aussi intéressé au modèle de déclenchement restreint, où on considère que seuls les événements appartenant à un ensemble des événements primaires qu'on note P peuvent engendrer des descendants. Le modèle ETAS

affirme que chaque événement engendre des descendants avec une fréquence proportionnelle à la variable K_i donnée dans l'équation (25). Ce qui suggère que seuls les séismes avec une magnitude importante peuvent engendrer des descendants. Ainsi l'ensemble des séismes qu'on appelle des événements primaires peut être l'ensemble des séismes avec une magnitude $m_i \geq M_s$ où M_s est une magnitude seuil.

12.4.2 Simulation

D'après l'approche de la section 12.2, on a implémenté le programme ci-dessous dans R pour simuler des processus ETAS avec l'intensité conditionnelle donnée dans l'équation (26). On rappelle qu'il s'agit d'une simulation basée sur l'algorithme d'amincissement d'Ogata et que les magnitudes sont simulées indépendamment du temps.

```

params=list(k=0.54,
           c=1.5,
           p=2,
           lambda=1.2,
           T=100,
           alpha=1,
           beta=1.6,
           M0=5,
           model="ETAS")

# Simulation des magnitudes par inversion de la fonction de rpartition
#
# @param params Nombre de paramtres
# @param n Nombre de tirages
#
# @return n ralisation de la magnitude
sim.magnitude=function(params,n){
  return(params$M0+rexp(n)/params$beta)
}

#Simulation du processus
#
# @param param List of parameters
sim.Hawkes.ETAS=function(params){
  t=c() # t est le vecteur des temps d'occurence du processus
  m=c() # m est le vecteur des magnitudes
  s=0 # temps d'occurence des vnements
  intensite=c() # intensite est le vecteur des intensits

  intensity . local<-function(s,t,m,params)

```



```

    {
      return(params$lambda+sum(params$k*exp(params$alpha*(m-params$M0))/(params$c+s-t)^para
    }
  }

# m et t sont des vecteurs
# s est un scalaire
lambda.sup=intensity.local(s,t,m,params) # calcul de l' intensit
while (s<params$T) {
  s=s+rexp(1,lambda.sup) # incrmte le prochain temps, exponentiel
  lambda.temp=intensity.local(s,t,m,params) # nouvelle intensit
  if (runif(1,0,1)<lambda.temp/lambda.sup){ # on accepte s avec la
    proba lambda.temp/lambda.sup
    t=c(t,s) # actualisation du vecteur des dates d'occurence
    mag=sim.magnitude(params,1) # actualisation de la nouvelle magnitude
    m=c(m,mag)
    intensite =c(intensite,lambda.temp)
    #cat(sprintf("s=%f\ tm=%f\ tl=%f\n",s,mag,lambda.temp))
  }
  lambda.sup=lambda.temp #on modifie lambda.sup chaque itration
}
idx<- t<=params$T
t<- t[idx]
m<- m[idx]
intensite <- intensite[idx]
result <-
  data.frame(nbpoints=1:length(t),time=t,magnitudes=m,intensities=intensite)
class(result) <- c("hawkes",class(result)) # cration d'une classe
hawkes
# Ajoute comme attribut les paramtres
attr(result,"params") <- params
# Ajoute comme attribut la fonction d' intensit
attr(result,"intensity") <- Vectorize(function(s) {
  idx<- (t<=s) # indices garder
  return(intensity.local(s,t[idx],m[idx],params))},"s")
return(result)
}

go <- function() {
  #commandes d'execution pour un exemple
  #set.seed(2020)
  #il faut que alpha<beta et k$beta/(alpha-beta)<(p-1)*c^(p-1)
  res<- sim.Hawkes.ETAS(params)
  plot(res$time,res$nbpoints,type="s",ylab="N(t)",xlab="time t",col="blue",
    main="Realization of Hawkes point process – ETAS model")
  return(invisible(res))
}

```

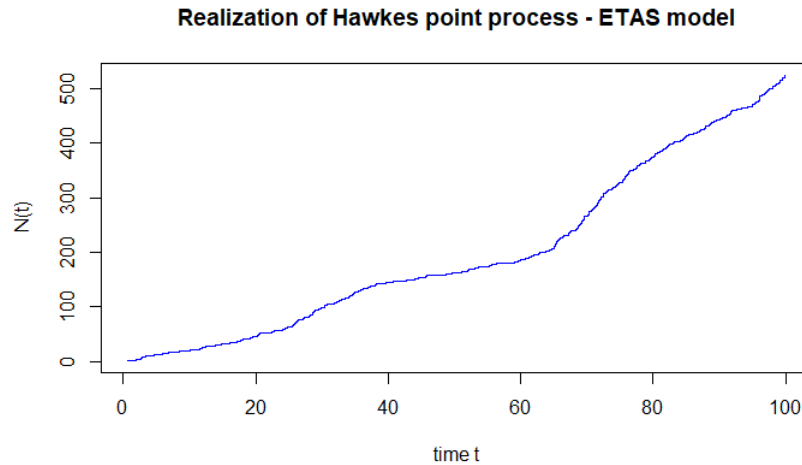


FIGURE 32 – échantillon de Hawkes simulé de paramètres $k = 0.54$, $c = 1.5$, $p = 2$, $\lambda = 2$, $\alpha = 1$ avec $T = 100$, $M_0 = 5$ et $\beta = 1.6$.

```
}
```

```
#commande d'exécution
res=go()
```

```
#on surcharge plot, pour la classe hawkes
#plot trace le nombre d'vnements avec l'intensité au dessus
plot.hawkes<-function(res){
  plot(res$time,res$nbpoints,type="s",ylab="",xlab="",
        main="")
  par(new = TRUE)
  plot(res$time,res$intensities ,type="l",xlab="",ylab="",axes=F,col="red",main="number
    of events and intensity")
  axis(side=4, at =
        pretty(range(res$intensities)),col="red",col.axis="red")
  mtext("time", side=1,line=2.5)
  mtext("number of events", side = 2,line=2.5,cex=0.8)
  #mtext("intensity", side = 4,line=2.5,col="red")
}
plot(res)
```

12.4.3 Inférence

La log vraisemblance s'écrit :

$$\ell = \sum_{i=1}^k \log(\lambda^*(t_i)) - \int_0^T \lambda^*(u) du,$$

et d'après l'équation (26)

$$\lambda^*(t) = \lambda + \sum_{t_i < t} K \frac{\exp(\alpha(m_i - M_0))}{(t - t_i + c)^p}.$$

Donc pour une suite de dates d'occurrences $(t_i)_{1 \leq i \leq k}$, on peut simplifier l'expression de cette fonction sur $[0, t_k]$:

$$\begin{aligned} \int_0^{t_k} \lambda^*(u) du &= \int_0^{t_1} \lambda^*(u) du + \int_{t_1}^{t_k} \lambda^*(u) du \\ &= \int_0^{t_1} \lambda du + \int_{t_1}^{t_k} \left(\lambda + \sum_{t_i < u} K \frac{\exp(\alpha(m_i - M_0))}{(u - t_i + c)^p} \right) du \\ &= \lambda t_k + \int_{t_1}^{t_k} \sum_{t_i < u} K \frac{\exp(\alpha(m_i - M_0))}{(u - t_i + c)^p} du. \end{aligned}$$

Chacun des t_i a une influence sur l'intensité sur l'intervalle $[t_i, t_k]$ donc

$$\begin{aligned} \int_0^{t_k} \lambda^*(u) du &= \lambda t_k + \sum_{i=1}^k \int_{t_i}^{t_k} K \frac{\exp(\alpha(m_i - M_0))}{(u - t_i + c)^p} du \\ &= \lambda t_k + \frac{K}{1-p} \sum_{i=1}^k \left(\exp(\alpha(m_i - M_0)) \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right] \right). \end{aligned} \quad (27)$$

Donc

$$\begin{aligned} \ell &= \sum_{i=1}^k \ln \left(\lambda + K \exp(\alpha(m_i - M_0)) \sum_{j=1}^{i-1} \frac{1}{(t_i - t_j + c)^p} \right) - \lambda t_k \\ &\quad - \frac{K}{1-p} \sum_{i=1}^k \left(\exp(\alpha(m_i - M_0)) \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right] \right). \end{aligned} \quad (28)$$

Le premier terme de cette log-vraisemblance est de complexité quadratique en fonction du nombre d'événements. Ce qui rend le calcul de cette fonction coûteux. Ogata(1993) [13] a proposé une méthode de calcul de la log-vraisemblance qui permet de se ramener à une complexité linéaire. Ici on garde l'expression (28).

Le programme ci-dessous donne le code R pour calculer cette fonction. Le code renvoie l'opposé de la log-vraisemblance.

```

log_likelihoood_ETAS=function(x,beta,t,m,M0){
  #x,t et m sont des vecteurs
  #beta et M0 sont des scalaires
  K=x[1];c=x[2];p=x[3];lambda=x[4];alpha=x[5]
  if (beta<alpha | K*beta/(beta-alpha)>(p-1)*c^(p-1)){return(0)}
  #valeur fixe 0 si les contraintes du modele ne sont pas respectes
  n=length(t)
  a1=rep(0,n)
  a2=rep(0,(n-1))
  for (i in 2:n){
    t_inf=t[t<t[i]]
    a1[i]=exp(alpha*(m[i]-M0))*sum(1/(c+t[i]-t_inf)^p)
    a2[i]=exp(alpha*(m[i]-M0))*((1/(c+t[n]-t[i])^(p-1))-(1/(c)^(p-1)))
  }
  l= sum(log(lambda+K*a1))-t[n]*lambda-sum(a2)*K/(1-p)
  return(-l)}

#Pour tracer la figure
##fonction pour trouver un k qui respecte les contraintes
#pour c,p,beta et alpha donns
find2_k=function(c,p,beta,alpha){
  return((p-1)*c^(p-1)*(beta-alpha)/beta-0.001)
}
# chantillon simul
params=list(k=0.54,
            c=1.5,
            p=2,
            lambda=1.2,
            T=100,
            alpha=1,
            beta=1.6,
            M0=5,
            model="ETAS")
res=sim.Hawkes.ETAS(params)
t=res$time
m=res$magnitudes

## trac de la log-vraisemblance
#en fonction de deux paramtres: ici p et alpha
#vecteurs des valeurs des deux paramtres
x <- seq(0.1,4,by=0.2) #p
y <- seq(0.1,4,by=0.2) #alpha

#matrice des valeurs de la log-vraisemblance
z=matrix(rep(0, length(x)*length(y)), length(x), length(y))

```

```

for(i in 1:length(x)){
  for (j in 1:length(y)){
    z[i,j]=log_likelihood_ETAS(c(params$k,params$c,x[i],params$lambda,y[j]),
    params$beta,t,m,params$M0)
  }
}
# trac en 3D
pmat<-persp(x,y,z, xlim = range(x), ylim = range(y),zlim = range(z, na.rm
= TRUE),xlab="p",ylab="alpha", zlab = "-lnL",theta = -20, phi =
20,scale=TRUE, axes=T, ticktype="detailed")

#On ajoute au graphe le point qui represente les vrais paramtres de
l'echantillon
e=c(params$k,params$c,params$p,params$lambda,params$alpha)

x0=params$p;y0=params$alpha
z0=log_likelihood_ETAS(e,params$beta,t,m,params$M0)
trans3d <- function(x,y,z, pmat) {
  tr <- cbind(x,y,z,1) %*% pmat
  list (x = tr[,1] /tr[,4], y= tr[,2] /tr[,4])
}
points(trans3d(x0,y0,z0,pmat),pch=16, col="red")

```

Les figures 33 et 34 montrent respectivement la log-vraisemblance d'un échantillon ETAS en fonction de k et λ et puis en fonction de c et p . Dans la littérature, on utilise la méthode de Davidon-Fletcher-Powell (1976) [6] pour optimiser la log-vraisemblance de l'équation (28).

12.4.4 Package SAPP

Nous avons utilisé le package SAPP de R pour faire l'estimation des paramètres par maximisation de la vraisemblance. La fonction `etasap` prend en argument le vecteur des temps d'occurrence, le vecteur des magnitudes des séismes ainsi que la magnitude seuil et la magnitude de référence qu'on considère égales, et renvoie les valeurs estimées des paramètres λ , k , c , p et α .

La procédure d'optimisation utilisée est celle de Davidon-Fletcher-Powell [4]. La fonction `etasap` présente deux approches d'estimation, la première est exacte en utilisant la log-vraisemblance de l'équation (28), la deuxième est approximative en utilisant une expression approximative de la log-vraisemblance pour laquelle la complexité de calcul est linéaire, ce qui permet de gagner du temps mais de perdre de la précision, il est possible de passer de l'approche exacte à l'approche approximative en modifiant le paramètre `approx` dans la fonction `etasap`, qui est égal par défaut au niveau 2 d'approximation. Le code ci-dessous donne le programme R pour réaliser cette estimation. Concrètement, pour estimer les paramètres il suffit d'appliquer la fonction `etasapp` au vecteur des temps

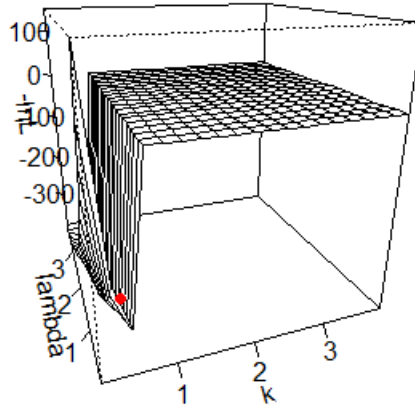


FIGURE 33 – la log-vraisemblance d'un échantillon ETAS en fonction des paramètres k et λ variant de 0.1 à 4. Les vrais paramètres sont représentés par le point rouge.

t , le vecteur des magnitudes m ainsi que la magnitude seuil M_0 et l'horizon du temps T . Cette fonction renvoie les valeurs estimées, l'opposé du maximum de la log-vraisemblance et la valeur du AIC divisée par 2.

```
library(SAPP)
#estimation
#@return liste des valeurs  estimes  des paramtres ETAS: lambda(not
    mu),k,c,alpha et p
estim_ETAS<-function(t,m,M0,T){
    #t et m sont des vecteurs
    #M0 et T sont des scalaires
    x=etasap(t,m,threshold=M0,reference=M0,parami=c(0.2,0.2,1,0.3,1.5),tstart=0,zte=T,plot=F)
    par=x$param
    return(par)}
par=estim_ETAS(t,m,params$M0,params$T) #liste des valeurs
```

Le code ci-dessous donne le programme R pour calculer les résidus de l'analyse résiduelle.

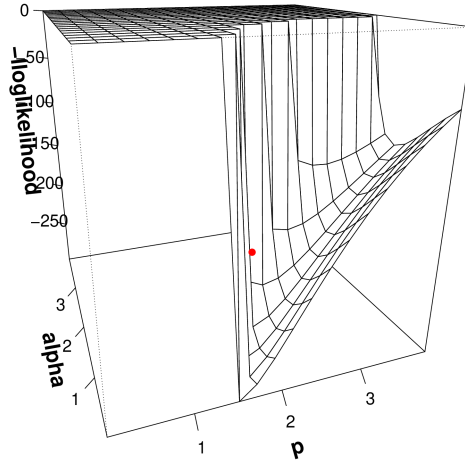


FIGURE 34 – la log-vraisemblance d'un échantillon ETAS en fonction des paramètres p et α variant de 0.1 à 4. Les vrais paramètres sont représentés par le point rouge.

```

#l' intensit cumule = integrale de l' intensit conditionnelle
#@return la valeur de l' intensit cumule un instant donn
int_cumul_ETAS=function(s,params,t,m){
  #s est l'instant auquel on calcule l' intensit cumule
  #t et m sont des vecteurs
  A=t[t<s]
  mag=m[1:length(A)]
  return(params$lambda*s+(params$k/(1-params$p))*sum(exp(params$alpha*(mag-params$M0))*((1/(pa
    }

#les rsidus = transformes des dates d'occurence par l' intensit cumule
#@return le vecteur des rsidus
residu_ETAS=function(t,m,params){
  res=rep(0,length(t))
  for (i in 2:length(res)){
    res[i]=int_cumul_ETAS(t[i],params,t,m)
  }
}

```

```
return(res))}
```

Le code ci-dessous donne le code R pour effectuer les procédures de l'analyse résiduelle.

```
##Test d'enveloppe: vrifier si les rsidus forment un processus de
Poisson homogne d' intensit 1
#trace l'enveloppe des processus de poisson homogne d' intensit 1
#superpose aux rsidus de plusieurs chantillons avec les valeurs
estimes comme leurs paramtres ETAS
#
# gnration d'un processus de poisson homogne avec n tirages
Poisson_homogene=function(T,n){
  #T et n sont des scalaires :horizon de temps et nombre de tirages
  i=2;t=0
  while (t[length(t)]<T){
    t[i]=t[i-1]+rexp(1,n)
    i=i+1
  }
  return(t) }
#test d'enveloppe
envelope_residuals_ETAS=function(n,params){
  #n est le nombre d' chantillons de rsidus tracs
  #params liste des paramtres ETAS
  process=sim.Hawkes.ETAS(params)
  v=process$time
  m=process$magnitudes
  N=process$nbpoints
  res=residu_ETAS(v,m,params)
  plot(res,N,type="s",col="green",xlab="",ylab="",main="enveloppe
test",cex.main=1.5)
  mtext(expression(paste(Lambda,"(days)")),
    side=1,font=2,line=2.5,cex=1.3)
  mtext("number of events", side = 2,line=2.5,font=2,cex=1.3)
  for (i in 1:n){ #on trace les processus de Poisson(1)
    par(new = T)
    v=Poisson_homogene(params$T,1)
    N=c(0:(length(v)-1))
    plot(v,N,type="s",col="green",axes=F,xlab="",ylab="")
  }
  for (i in 1:n){ #on trace les residus des processs de Hawkes simulés
    par(new = T)
    process=sim.Hawkes.ETAS(params)
    v=process$time
    m=process$magnitudes
    N=process$nbpoints
```



```

    res=residu_ETAS(v,m,params)
    plot(res,N,type="s",col="red",axes=F,xlab="",ylab="")
  }}
#commande d'xcution
#avec les valeurs de la liste params

envelope_residuals_ETAS(100,params)
##Test de Kolmogrov Smirnov
# vrifier si les interarives des rsidus suivent une loi exponentielle
# de paramtre 1
#
#@return p-valeur de kolmogrov et trace les fonctions de rpartition du
# test
res_kolmogrov<-function(t,m,params){
  res=residu_ETAS(t,m,params)
  #les interarives des rsidus du vecteur t
  diff_res=c(res[1],diff(res))
  #p-vaeur du test de Kolmogrov Smirnov
  p_val=ks.test(diff_res,"pexp",1)
  #distance de kolmogrov entre les fonctions de repartition
  D=p_val$statistic
  inter=sort(diff_res)
  plot(ecdf(inter),ylab="",xlab="",axes=F,main = 'Cumluative
    Distribution functions',pch=20,cex.main=1.5)
  axis(1)
  axis(2)
  mtext(expression(paste(Lambda,"(days)")),
    side=1,line=2.5,font=2,cex=1.3)
  #legend("bottomright",legend=c("empirical function", "theoretical
    function"),col=c("black", "red"), lty=1,1,cex=0.8)
  mtext(paste("KS Distance=",
    round(D,2),"p-value=",round(p_val$p.value,2)),side=3,cex=1.2)
  lines(inter, pexp(inter,1),ylab="",col="red",type="l")
  points(3, 0.1, pch = 20,col="red", lwd = 2, cex = 1.2)
  text(3.2, 0.1, "theoretical function", cex = 1.2, font = 1, adj = 0)
  points(3, 0.2, pch = 20, lwd = 2, cex = 1.2)
  text(3.2, 0.2, "empirical function", cex = 1.2, font = 1, adj = 0)
  return(invisible(p_val$p.value))
}
#commande d'xcution
res_kolmogrov(t,m,params)

###Test de rsidus de martingale
#residus de martingale a l'instant s=nombre d'vnements arrivs
# jusqu' s= intensit cumule (s)

```

```

#on verifie si les interarrives des rsidus ascillent autour de 0
residus__ETAS__mart=function(ind,t,m,params){
  #ind est le vecteur des indices des points pris pour la grilles des
  rsidus
  resid=rep(0,length(ind))
  for (i in 2:length(resid)){
    #il y a ind[i] vnements arrivs jusqu' t[ind[i]]
    resid[i]=ind[i]-int__cumul__ETAS(t[ind[i]],params,t,m) #rsidu
    l'instant t[ind[i]]
  }
  return(resid)
}
residus__martingale__ETAS=function(t,m,params){
  n=200
  #ind vecteur des indices des points pris pour la grilles des rsidus
  ind=seq(1,length(t),by=floor(length(t)/n))
  #z vecteur des dates des rsidus
  z=rep(0,length(ind))
  for (i in 1:length(z)){
    z[i]=t[ind[i]]
  }
  res__z=residus__ETAS__mart(ind,t,m,params)
  #liste des differences
  diff__z=c(res__z[1],diff(res__z))
  #graphe des residus de martingale
  plot(z, diff__z,axes=F,xlab="",ylab="",main = 'Martingale
  residuals',type='l',col=2,cex.main=1.5)
  axis(1)
  axis(2)
  mtext("days", font=2,side=1,line=2.5,cex=1.3)
  mtext("residuals", font=2,side = 2,line=2.5,cex=1.3)
}
#commande d'xecution
residus__martingale__ETAS(t,m,params)

#intervalle de confiance
n=200
#ind vecteur des indices des points pris pour la grilles des rsidus
ind=seq(1,length(t),by=floor(length(t)/n))
#z vecteur des dates des rsidus

resid=rep(0,length(ind))
for (i in 2:length(resid)){
  #il y a ind[i] vnements arrivs jusqu' t[ind[i]]
  resid[i]=int__cumul__ETAS(t[ind[i]],params,t,m) #rsidu l'instant

```

```

      t[ind[i]]
    }
    #liste des differences
    diff_res=c(-res_z[1],-diff(res_z))
    intervalle_c(diff_res)

    #Pour tracer l' intensit avec lesgraphes des rsidus
    #estimation des paramtres
    betaes=estim_beta(m,params$M0)
    par=estim_ETAS(t,m,params$M0,params$T) #liste des valeurs estimes
    lambdaes=par$mu;kes=par$K;ces=par$c;alphaes=par$alpha;pes=
    params_est=list(k=kes,
                    c=ces,
                    p=pes,
                    lambda=lambdaes,
                    T=params$T,
                    alpha=alphaes,
                    beta=betaes,
                    M0=params$M0,
                    model="ETAS")

    params_est
    #pour tracer l' intensit superpose au nombre d' vnements
    plot.hawkes<-function(res){
      plot(res$time,res$nbpoints,type="s",ylab="",xlab="",
            main="")
      par(new = TRUE)
      plot(res$time,res$intensities ,type="l",xlab="",ylab="",axes=F,col="red",main="number
            of events and intensity",cex.main=1.6)
      axis(side=4, at =
            pretty(range(res$intensities)),col="red",col.axis="red")
      mtext("days",font=2,side=1,line=2.5,cex=1.3)
      mtext("number of events", side = 2,line=2.5,font=2,cex=1.3)
      #mtext("intensity", side = 4,line=2.5,col="red")
    }

    #commande d'execution
    #Graphes de l'analyse des rsidus
    par(mfrow=c(2,2))
    plot(res)
    envelope_residuals_ETAS(100,params_est)
    res_kolmogrov(t,m,params_est)
    residus_martingale_ETAS(t,m,params_est)

```

Le code ci-dessous donne le code R pour tracer les intervalles de confiance des r s dus de martingale pour les 3 mod les.

```

#bande de confiance
conf.band<-function(t,intcum){
  #intcum vecteur des residus=image de t par int cumule
  #diff(intcum) les rsidus Yi
  mart=cumsum(c(0,diff(intcum)))
  nb=c(1:length(t))
  #fonction sui calcule le seuil
  seuil<-function(N){
    return(qgamma(0.05,shape=N,scale=1,lower.tail = F))
  }
  seuil_low<-function(N){
    return(qgamma(0.95,shape=N,scale=1,lower.tail = F))
  }
  s_up<-Vectorize(seuil,vectorize.args = "N")
  s_low<-Vectorize(seuil_low,vectorize.args = "N")
  #seuil suprieur N par N
  up=s_up(nb)
  low=s_low(nb)
  result <- data.frame(martingale=mart,upper=up,lower=low)
  return( invisible ( result ))
}

```

```

#Pour le modele exponentiel

a=0.6;b=0.8;lambda=1.2;T=100;
res=sim.Hawkes.Ogata(a,b,lambda,T)
t=res$time
#estimation
est=estimation(t)
lambda=est[1];a=est[2];b=est[3]
length(t)
lambda
int_cumul=function(s,lambda,a,b,temps){
  A=temps[temps<s]
  return(lambda*s-(a/b)*sum(exp(-b*(s-A))-1)) }

int_cum_exp<-Vectorize(int_cumul,vectorize.args = "s")
intcum=int_cum_exp(t,lambda,a,b,t)
vect<-conf.band(t,intcum)
dev.off()
plot(vect$mart,type="l",xlab="",ylab="")
mtext(expression(sum(Y[i])),line=2.5,cex=1.4,side=2)
lines(vect$upper,col="red")
mtext("Number of events",side=1,line=2.5,cex=1.4)
lines(vect$lower,col="green")

```

```

#legende: a modifier l'emplacement selon le graphe
points(250, 30, pch = 20,lwd = 2, cex = 1.2)
text(260, 30, "martingale", cex = 1.2, font = 1, adj = 0)
points(250, 50, pch = 20,col="red", lwd = 2, cex = 1.2)
text(260, 50, "upper bound", cex = 1.2, font = 1, adj = 0)
points(250, 70, pch = 20, col="green",lwd = 2, cex = 1.2)
text(260, 70, "lower bound", cex = 1.2, font = 1, adj = 0)

#Pour ETAS

params=list(k=0.54,
            c=1.5,
            p=2,
            lambda=1.2,
            T=100,
            alpha=1,
            beta=1.6,
            M0=5,
            model="ETAS")

res=sim.Hawkes.ETAS(params)
t=res$time
m=res$magnitudes

int_cumul_ETAS=function(s,params,t,m){
  #s est l'instant auquel on calcule l'intensité cumule
  #t et m sont des vecteurs
  A=t[t<s]
  mag=m[1:length(A)]
  return(params$lambda*s+(params$k/(1-params$p))*sum(exp(params$alpha*(mag-params$M0))*((1/(pa
    }

int_cum_f<-Vectorize(int_cumul_ETAS,vectorize.args = "s")
intcum=int_cum_f(t,params_est,t,m)
vect<-conf.band(t,intcum)
dev.off()
plot(vect$mart,type="l",xlab="",ylab="")
mtext(expression(sum(Y[i])),line=2.5,cex=1.4,side=2)
lines(vect$upper,col="red")
mtext("Number of events",line=2.5,cex=1.4,side=1)
lines(vect$lower,col="green")

#Pour le modele puissance
params=list(k=0.54,
            c=1.5,
            p=2,

```

```

        lambda=1.2,
        T=100,
        model="Puissance")
res=sim.hawkes.puissance(params)
t=res$time
int_cumul_puissance=function(temps,s,params){
  A=temps[temps<s]
  return(params$lambda*s+(params$k/(1-params$p))*sum(1/((params$c+s-A)^(params$p-1))-(1/param
    })

int_cum_puiss<-Vectorize(int_cumul_puissance,vectorize.args = "s")
intcum=int_cum_puiss(t,t,params)
mart=cumsum(c(0,diff(intcum)))
vect<-conf.band(t,intcum)
plot(vect$mart,type="l",xlab="",ylab="")
mtext(expression(sum(Y[i])),line=2.5,cex=1.4,side=2)
lines(vect$upper,col="red")
mtext("N",side=1)
lines(vect$lower,col="green")

###test d'enveloppe

int_cumul_puissance=function(temps,s,params){
  A=temps[temps<s]
  return(params$lambda*s+(params$k/(1-params$p))*sum(1/((params$c+s-A)^(params$p-1))-(1/param
    })
int_cum_puiss<-Vectorize(int_cumul_puissance,vectorize.args = "s")

#legende: a modifier l'emplacement selon le graphe
points(1500, 500, pch = 20,lwd = 2, cex = 1.2)
text(1550, 500, "martingale", cex = 1.2, font = 1, adj = 0)
points(1500, 200, pch = 20,col="red", lwd = 2, cex = 1.2)
text(1550, 200, "upper bound", cex = 1.2, font = 1, adj = 0)
points(1500, 350, pch = 20, col="green",lwd = 2, cex = 1.2)
text(1550, 350, "lower bound", cex = 1.2, font = 1, adj = 0)

```

12.4.5 Validation du modèle

Pour valider le modèle, nous effectuons une analyse résiduelle. L'intensité cumulée s'écrit d'après (27) :

$$\begin{aligned}
\Lambda(t) &= \int_0^t \lambda^*(s) ds = \int_0^{t_k} \lambda^*(s) ds + \int_{t_k}^t \lambda^*(s) ds \\
&= \lambda t_k + \frac{K}{1-p} \sum_{i=1}^k \left(\exp(\alpha(m_i - M_0)) \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right] \right) \\
&\quad + \int_{t_k}^t \left(\lambda + \sum_{i=1}^k \frac{\exp(\alpha(m_i - M_0))}{(c + s - t_i)^{p-1}} \right) ds \\
&= \lambda t + \frac{K}{1-p} \sum_{i=1}^k \left(\exp(\alpha(m_i - M_0)) \left[\frac{1}{(c + t_k - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right] \right) \\
&\quad + \frac{K}{1-p} \sum_{i=1}^k \left(\exp(\alpha(m_i - M_0)) \left[\frac{1}{(c + t - t_i)^{p-1}} - \frac{1}{(c + t_k - t_i)^{p-1}} \right] \right) \\
&= \lambda t + \frac{K}{1-p} \sum_{i=1}^k \left(\exp(\alpha(m_i - M_0)) \left[\frac{1}{(c + t - t_i)^{p-1}} - \frac{1}{c^{p-1}} \right] \right)
\end{aligned}$$

12.4.6 Application à des données simulées

On a appliqué le modèle ETAS à un échantillon de Hawkes simulé de paramètres $k = 0.54, c = 1.5, p = 2, \lambda = 2, \alpha = 1$ et $\beta = 1.6$ avec $T = 100, M_0 = 5$. L'estimation du paramètre β à partir des magnitudes a donné

$$\hat{\beta} = 1.64,$$

l'estimation des autres paramètres par la fonction `etasap` du package SAPP a donné :

$$\hat{k} = 1.75, \hat{c} = 1.87, \hat{p} = 2.34, \hat{\lambda} = 0.9, \hat{\alpha} = 0.42.$$

Ces valeurs sont proches des vrais paramètres. La figure 35 montre le résultat de l'analyse résiduelle. On observe que le test d'enveloppe, valide le modèle puisque les résidus tracés en rouge sont bien dans l'enveloppe de Poisson. Le test de Kolmogorov valide aussi le modèle car il renvoie une p -valeur égale à 0.89, ce qui permet de garder l'hypothèse nulle, et les fonctions de répartition sont proches. Le test des résidus de martingale valide aussi le modèle vu que les résidus oscillent autour de 0. La figure 36 montre le tracé de la martingale de ces résidus, $\sum Y_i$ de l'équation 19, avec une bande de confiance. On remarque la martingale est bien dans la bande de confiance.

12.4.7 Application à des données réelles

On applique le modèle ETAS aux données sismiques de la Guadeloupe. La moyenne des magnitudes des séismes est 2.31. Ainsi on choisit comme magnitude

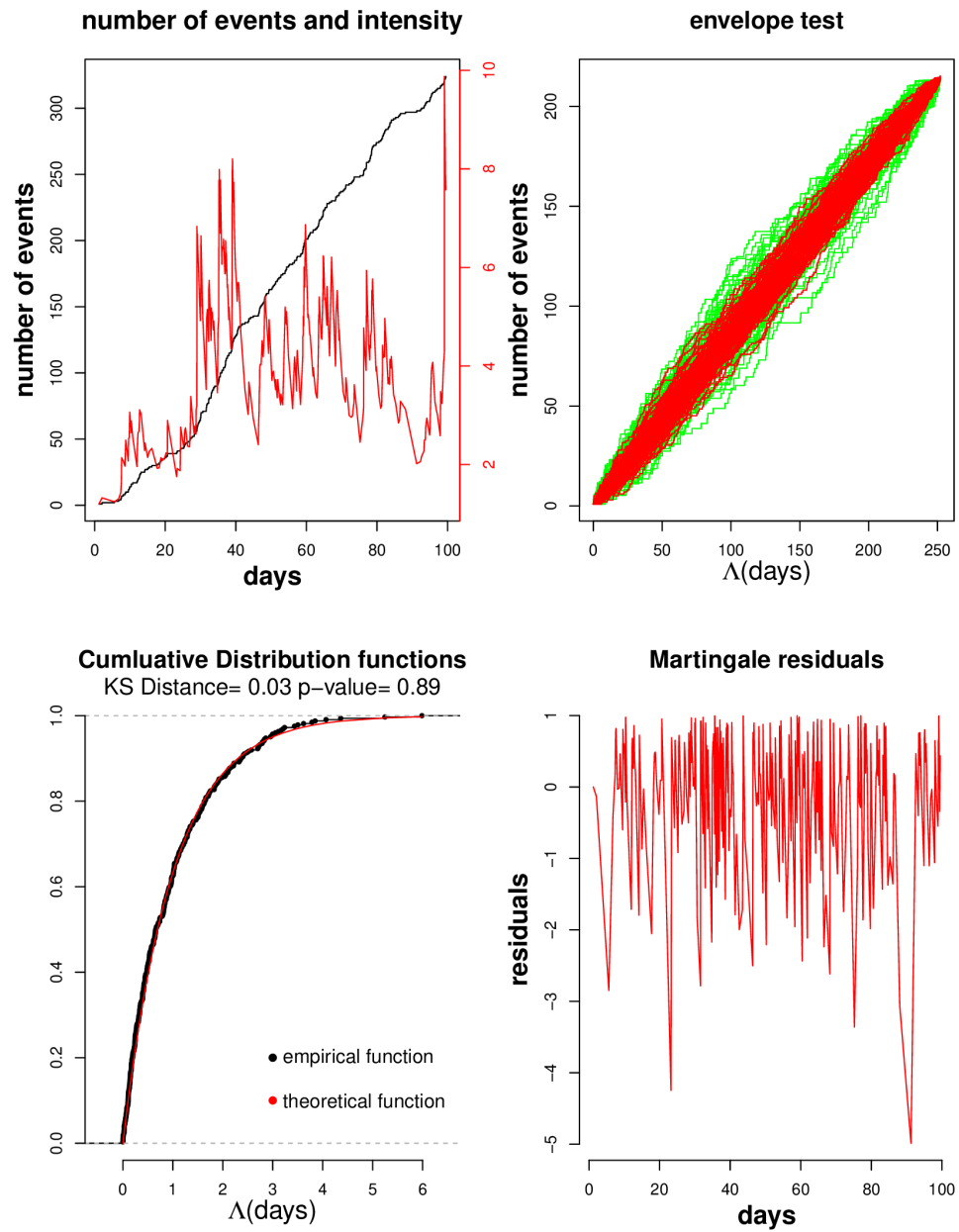


FIGURE 35 – analyse résiduelle d'un échantillon de Hawkes simulé selon le modèle ETAS

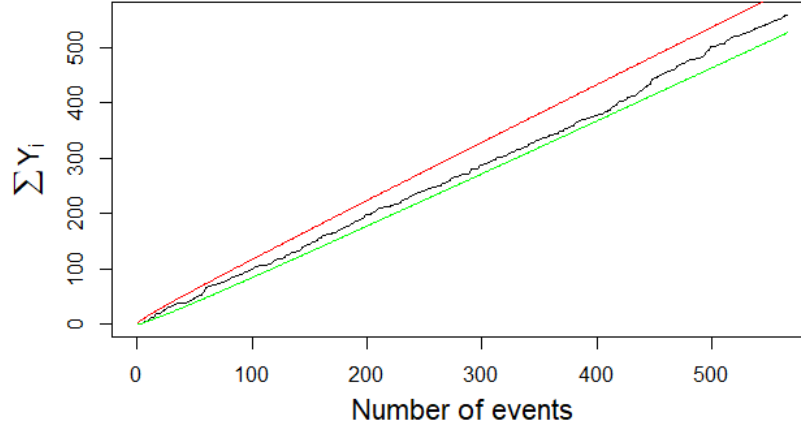


FIGURE 36 – Martingale des résidus et sa bande de confiance pour des données simulées selon le modèle ETAS.

seuil $M_0 = 2$. On passe à des données avec 1473 séismes. L'estimation de β à partir des magnitudes renvoie

$$\hat{\beta} = 1.67,$$

l'estimation des autres paramètres en utilisant la fonction `etasap` du package SAPP donne

$$\hat{k} = 0.38, \hat{c} = 0.67, \hat{p} = 3.25, \hat{\alpha} = 0.73, \hat{\lambda} = 0.97.$$

On peut interpréter la valeur de λ en disant qu'en moyenne, il y a 0.97 nouveaux séismes qui arrivent indépendamment des autres. On remarque que cette valeur est supérieure à celle qu'on a trouvé pour le modèle puissance et inférieure à la valeur du modèle exponentiel, à savoir respectivement 0.48 et 1.76.

La figure 37 donne le résultat de l'analyse résiduelle. Pour le test d'enveloppe, on voit que les résidus tracés en rouge sont globalement dans l'enveloppe de Poisson. Pour le test de Kolmogorov, les fonctions de répartition théorique et empirique sont proches, de plus il renvoie une p -valeur égale à 0.44 ce qui permet de garder l'hypothèse de résidus exponentiels, donc ce test valide le modèle. Concernant, le test de résidus de martingale, ces derniers oscillent autour de 0 mais s'éloignent légèrement à partir du jour 0. Ces résidus ont une allure similaire à celle qu'on a obtenu pour le modèle exponentiel, ils ont notamment le même pic aux alentours du jour 230, cependant les résidus ETAS s'éloignent moins de 0 que les résidus du modèle exponentiel. La figure ?? montre le tracé de la martingale de ces résidus, $\sum Y_i$ de l'équation 38, avec une bande de confiance. On remarque la martingale est au dessous de la borne supérieure de la bande mais

elle sort légèrement de la bande inférieure pour certaines valeurs. Ce résultat est nettement mieux que celui obtenu avec le modèle exponentiel.

13 Comparaison des modèles

On a travaillé sur les processus de Hawkes selon 3 modèles : le modèle exponentiel, le modèle puissance et le modèle ETAS pour le cas marqué. Dans cette section on compare les résultats trouvés pour chacun de ces modèles appliqués aux données sismiques de la Guadeloupe, en comparant les résultats de l'analyse résiduelle dans un premier temps et en comparant les modèles selon le critère d'Akaike dans un deuxième temps. Les données utilisées dans le modèle ETAS ne correspondent qu'aux séismes avec une magnitude supérieure à 2, contrairement aux modèles exponentiel et puissance qu'on a appliqué à la totalité des données. Ainsi pour effectuer une comparaison selon le critère d'Akaike, on applique les 3 modèles aux données utilisées pour le modèle ETAS pour avoir des données de même taille.

13.1 Modèle exponentiel

Le modèle exponentiel est défini par son intensité conditionnelle qui s'écrit :

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} \alpha \exp(-\beta(t_k - t)).$$

L'application de ce modèle aux données sismiques a renvoyé les valeurs suivantes :

$$\hat{\lambda} = 1.76, \hat{\alpha} = 3.44, \hat{\beta} = 5.06.$$

La figure 39 montre que l'analyse résiduelle de ce modèle a été validée par le test d'enveloppe. Les fonctions de répartition du test de Kolmogorov-Smirnov étaient proches, mais la p -valeur était faible. Concernant les résidus de martingale, ils s'éloignaient de 0 à partir d'un certain rang. L'application du modèle exponentiel aux données ayant une magnitude supérieure à 2 renvoie les valeurs suivantes :

$$\hat{\lambda} = 1.06, \hat{\alpha} = 3.06, \hat{\beta} = 4.31.$$

La figure 40 montre le résultat de l'analyse résiduelle appliqué à ce modèle avec ces données. On remarque que les résultats sont améliorés. La p -valeur de Kolmogorov pour ces données est égale à 0.37 ce qui permet de garder l'hypothèse des résidus exponentiels. Les résidus de martingale s'éloignent moins de 0 par rapport aux résidus de la figure 39. On peut justifier cette amélioration par le fait que les séismes avec une magnitude importante sont les plus susceptibles d'engendrer des descendants selon le modèle de branchement de Hawkes. La valeur du critère AIC est :

$$\text{AIC}_{\text{exponentiel}} = -2629.13.$$

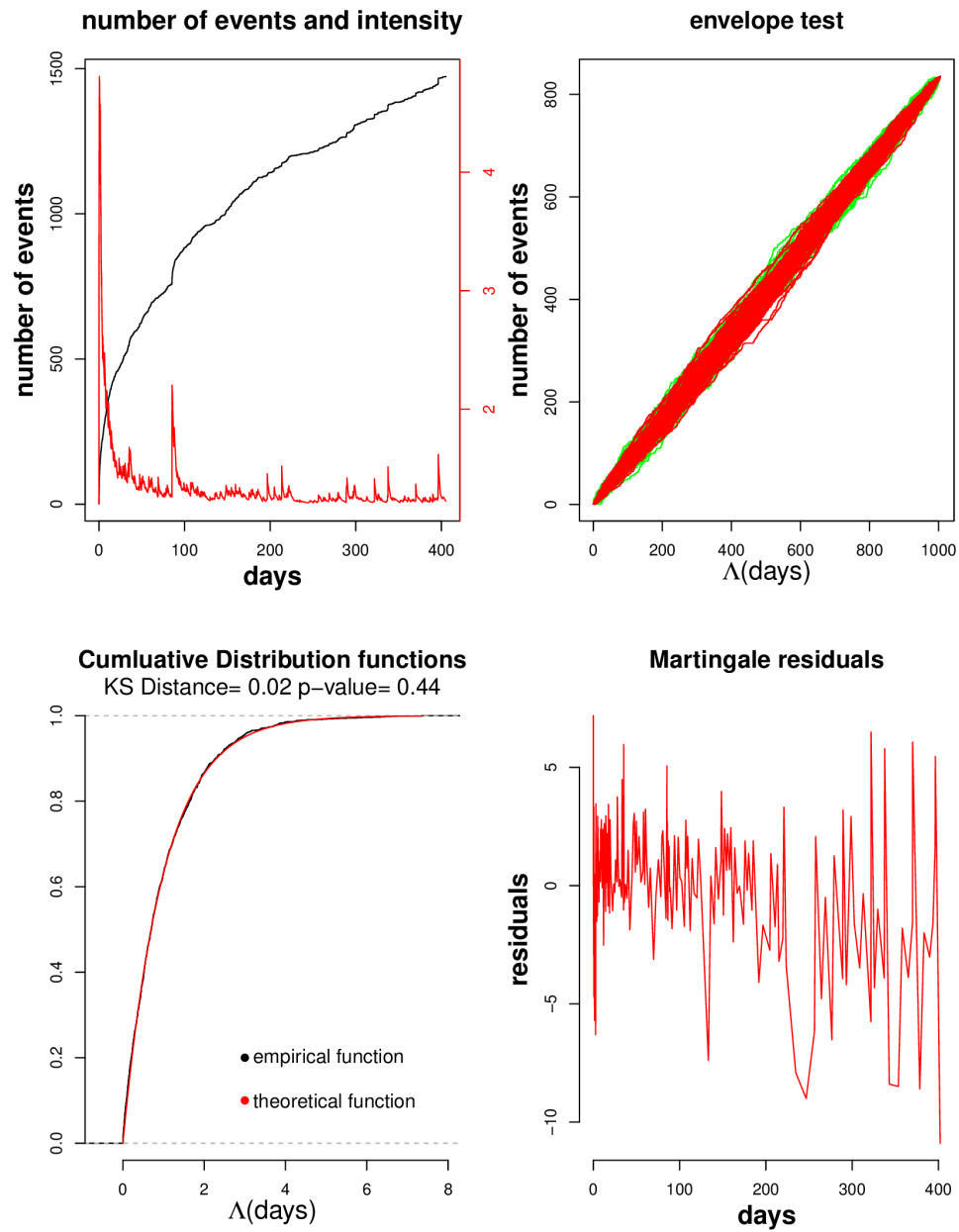


FIGURE 37 – Analyse résiduelle appliquée aux données sismiques réelles selon le modèle ETAS.

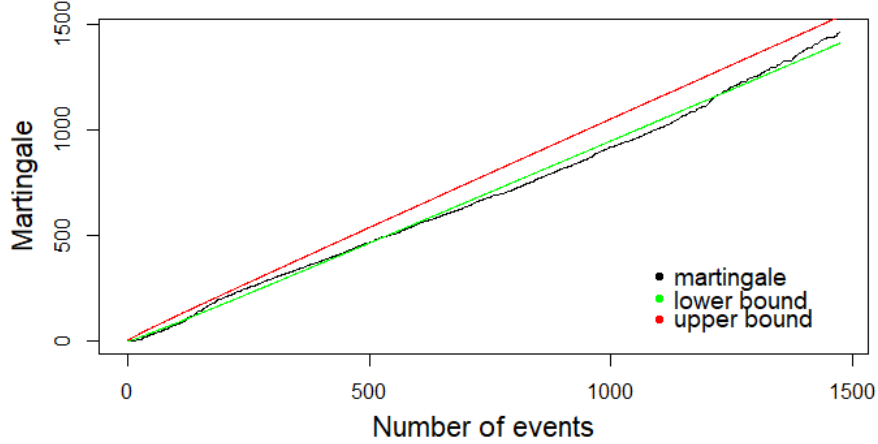


FIGURE 38 – Martingale des résidus et sa bande de confiance pour les données sismiques selon le modèle ETAS.

13.2 Modèle puissance

L'intensité conditionnelle du modèle puissance s'écrit :

$$\lambda^*(t) = \lambda + \sum_{k, t_k < t} \frac{K}{(c + t - t_k)^p}.$$

L'estimation des paramètres des données sismiques selon ce modèle donne

$$\hat{k} = 1.15, \hat{c} = 1, \hat{p} = 2.25, \hat{\lambda} = 0.48.$$

La figure 30 montre le résultat de l'analyse résiduelle. Les résidus du test d'enveloppe sont globalement dans l'enveloppe de Poisson mais sortent un peu de cette enveloppe pour certains échantillons. Pour le test de Kolmogorov-Smirnov les fonctions de répartition théorique et empirique sont proches mais la p-valeur très petite. Le test des résidus de martingale quant à lui valide le modèle, car les résidus oscillent autour de 0.

L'application du modèle puissance aux données ayant une magnitude supérieure à 2 renvoie les valeurs suivantes :

$$\hat{k} = 0.44, \hat{c} = 0.5, \hat{p} = 2.25, \hat{\lambda} = 0.56.$$

La figure 40 montre le résultat de l'analyse résiduelle appliqué à ce modèle avec ces données. Ce résultat est similaire à celui obtenu pour les données complètes. La valeur du critère AIC est :

$$\text{AIC}_{\text{puissance}} = -2664.84.$$

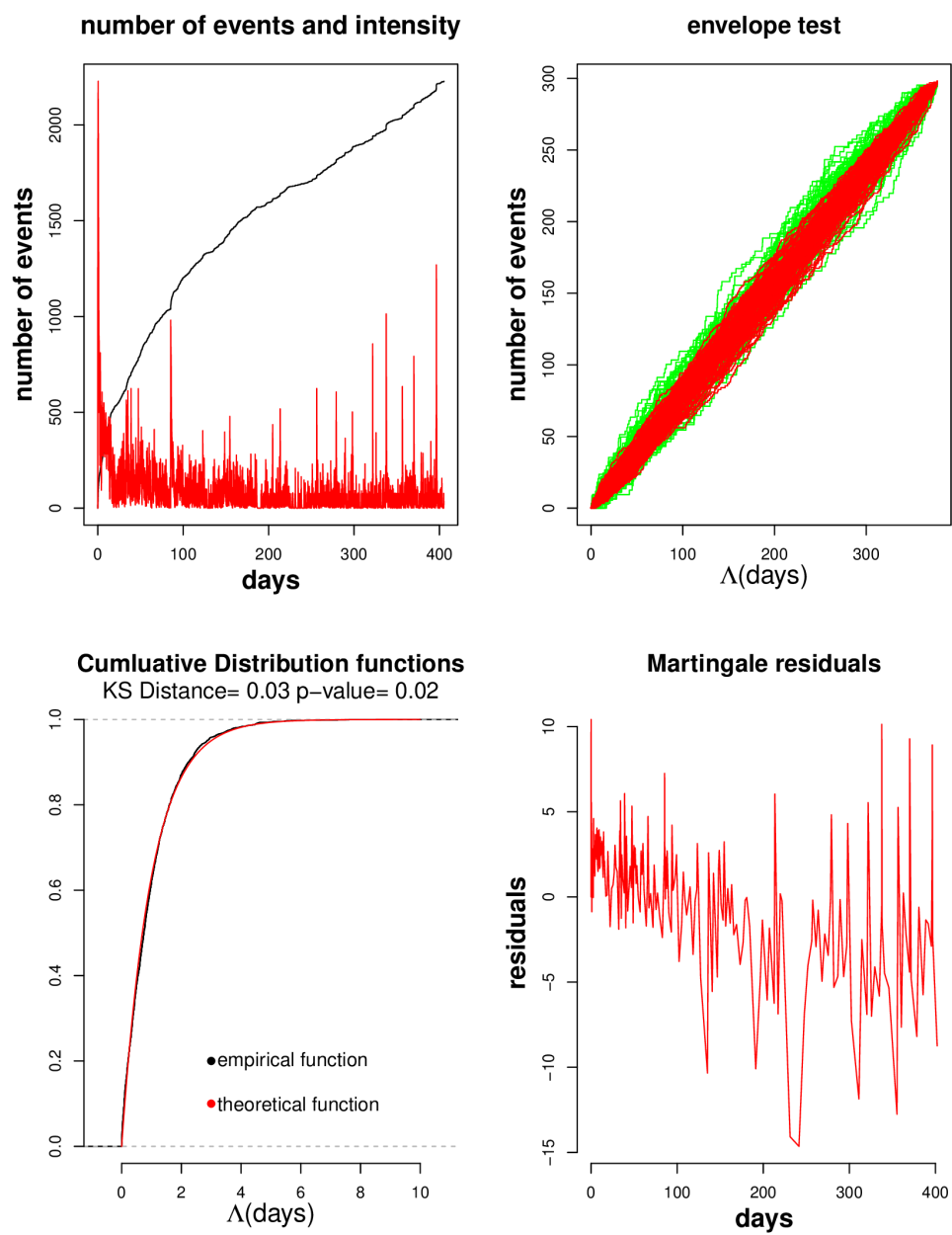


FIGURE 39 – Analyse résiduelle du modèle exponentiel appliqué au données sismiques

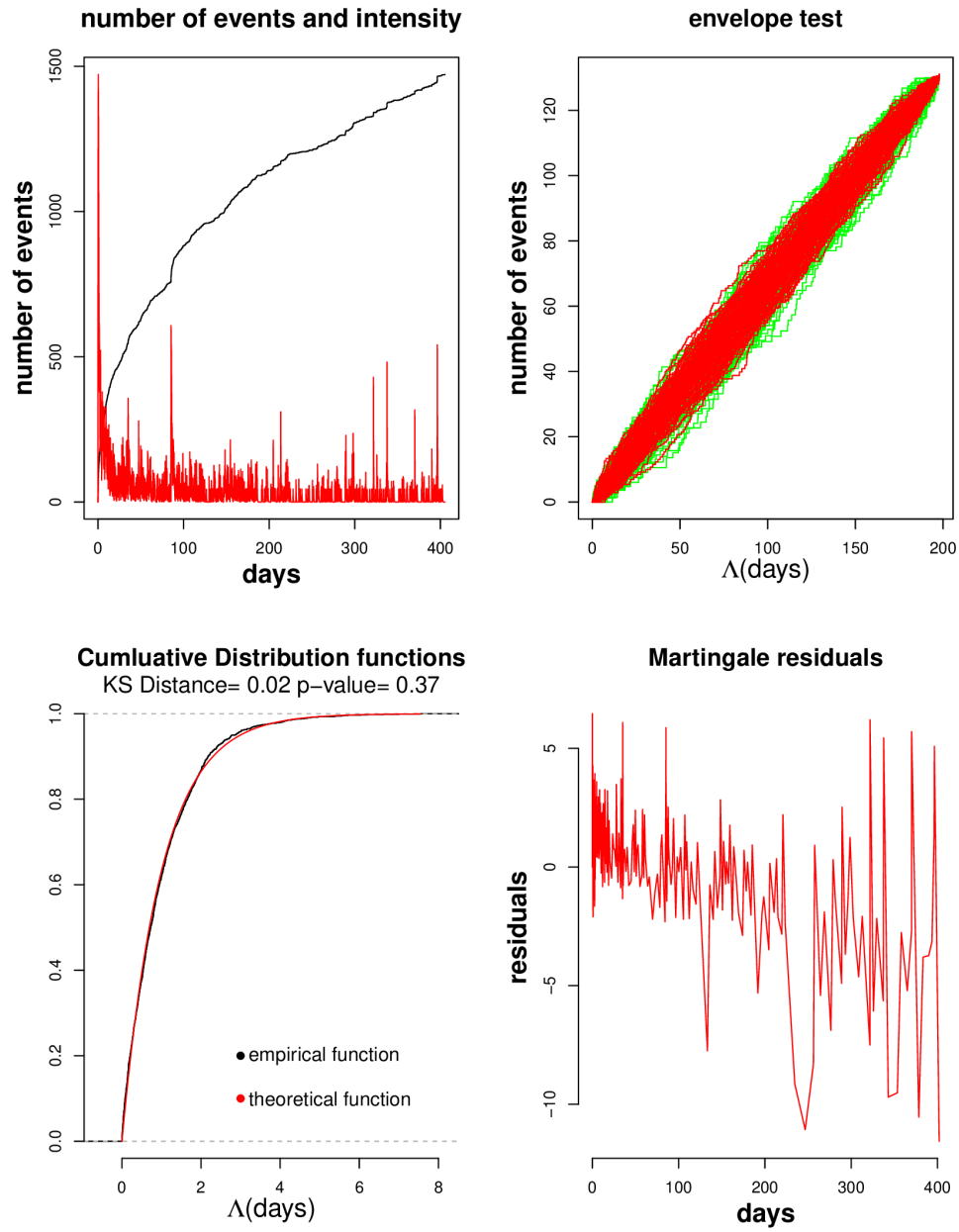


FIGURE 40 – Analyse résiduelle du modèle exponentiel appliqué au données sismiques ayant une magnitude $M \geq 2$

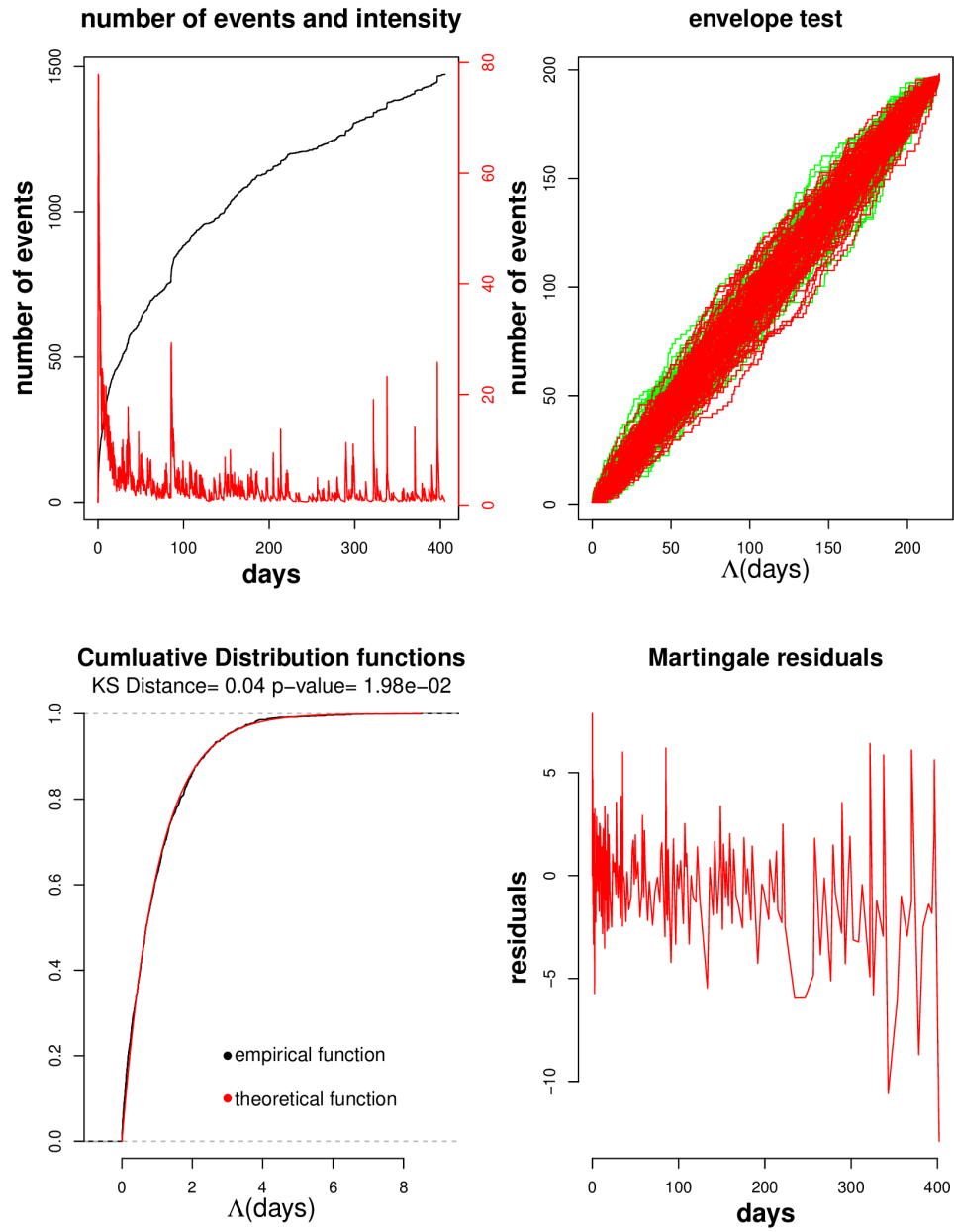


FIGURE 41 – Analyse résiduelle du modèle puissance appliqué au données sismiques ayant une magnitude $M \geq 2$

13.3 Modèle ETAS

Le modèle ETAS est un cas de processus de Hawkes marqués ayant comme intensité conditionnelle

$$\lambda^*(t) = \lambda + \sum_{t_i < t} K \frac{\exp(\alpha(m_i - M_0))}{(t - t_i + c)^p}.$$

L'estimation des paramètres des données sismiques selon ce modèle donne

$$\hat{\beta} = 1.67, \hat{k} = 0.38, \hat{c} = 0.67, \hat{p} = 3.25, \hat{\alpha} = 0.73, \hat{\lambda} = 0.97.$$

La figure 37 montre le résultat de l'analyse des résidus. Les résidus du test d'enveloppe sont globalement dans l'enveloppe de Poisson. Pour le test de Kolmogorov, les fonctions de répartition théorique et empirique sont proches, et la p-valeur égale à 0.44 ce qui valide le modèle. Concernant, le test de résidus de martingale, ces derniers oscillent autour de 0 mais s'éloignent légèrement à partir du jour 0. Ils ont une allure similaire à celle qu'on a obtenu pour le modèle exponentiel, ils ont notamment le même pic aux alentours du jour 230, cependant les résidus ETAS s'éloignent moins de 0 que les résidus du modèle exponentiel.

La valeur du critère AIC :

$$\text{AIC}_{\text{ETAS}} = -2778.51.$$

13.4 Conclusion

Quand applique les modèles exponentiel et puissance aux données complètes, et le modèle ETAS aux données ayant une magnitude supérieure à 2, on observe que le test d'enveloppe est validé pour les 3 modèles, un peu moins pour le modèle puissance. Les fonctions de répartition du test de Kolmogorov-Smirnov sont proches dans les 3 cas mais seul le modèle ETAS renvoie une p-valeur grande qui permet de garder l'hypothèse de résidus exponentiels. Concernant les résidus de martingale, ils s'éloignent le plus de 0 pour le modèle exponentiel suivi par le modèle ETAS et puis le modèle puissance pour lequel ils oscillent bien autour de 0. Ainsi on remarque que le modèle ETAS est bien meilleur que les deux autres puisqu'il est bien validé par le test d'enveloppe et de Kolmogorov-Smirnov, et il renvoie un résultat meilleur pour les résidus de martingale par rapport au modèle exponentiel, concernant le modèle puissance, il n'est pas bien validé par le test d'enveloppe et le test de Kolmogorov-Smirnov.

Quand applique les 3 modèles aux données ayant une magnitude supérieure à 2, la performance du modèle puissance est la même. Le modèle exponentiel donne des résultats meilleurs, il est validé par le test d'enveloppe et par le test de Kolmogorov, de plus, les résidus de martingale s'éloignent moins de 0. Ces résultats sont similaires à ceux obtenus avec le modèle ETAS. Pour trancher on procède à une comparaison des modèles selon le critère d'Akaike. On a

$$\text{AIC}_{\text{ETAS}} < \text{AIC}_{\text{puissance}} < \text{AIC}_{\text{exponentiel}},$$

ce qui montre que le modèle ETAS est le meilleur suivi du modèle puissance et puis du modèle exponentiel. Le tableau 42 synthétise le résultat de l'appli-

Paramètres/ Modèles	λ	α	β	k	c	p	α ETAS	β ETAS	AIC
Exponentiel	1.06	3.06	4.31	—	—	—	—	—	-2629.13
Puissance	0.56	—	—	0.44	0.5	2.25	—	—	-2664.84
ETAS	0.97	—	—	0.38	0.67	3.25	0.73	1.67	-2778.51

FIGURE 42 – résultats de l'application des 3 modèles aux données sismiques avec des magnitudes supérieures à 2.

tion des 3 modèles aux données sismiques avec des magnitudes supérieures à 2.

14 Processus de Hawkes spatio-temporels

Le modèle de Hawkes spatio-temporel est une extension des processus de Hawkes temporels à la dimension spatiale. Cette répartition permet de mieux comprendre le processus de génération des événements et d'améliorer la prévision. Ainsi on s'intéresse aux dates d'occurrence des événements et à leurs localisations dans un ensemble $X \subset \mathbb{R}^d$.

14.1 Modélisation

Dans cette partie, on s'intéresse à la modélisation mathématique des processus de Hawkes spatio-temporels. On suivra une approche similaire à celle des processus de Hawkes temporels en se basant sur le travail d'Ogata (1998) [11].

Définition 9. L'intensité conditionnelle d'un processus spatio-temporel est définie par

$$\lambda^*(s, t) = \lim_{\Delta s, \Delta t \rightarrow 0} \frac{\mathbb{E}[N(B(s, \Delta s) \times [t, t + \Delta t)) | H_t]}{|B(s, \Delta s)| \Delta t},$$

où $|B(s, \Delta s)|$ est la mesure de Lebesgue de la boule $B(s, \Delta s)$ de rayon Δs et $N(A)$ est la mesure de comptage du nombre d'événements sur un ensemble $A \in X \times [0, T)$.

Définition 10. Un processus de Hawkes spatio-temporel est défini par son intensité conditionnelle qui s'écrit sous la forme :

$$\lambda^*(s, t | H_t) = \mu(s) + \sum_{t_i < t} g(s - s_i, t - t_i),$$

tels que les s_i représentent les localisations des événements, et les t_i représentent leurs dates d'occurrence. g est la fonction d'excitation, qui est positive.

Remarque 10. — Par souci de simplicité, on prend souvent une fonction d'excitation séparable dans le temps et l'espace telle que $g(s - s_i, t - t_i) = f(s - s_i)h(t - t_i)$.

— En particulier pour des locations (x, y)

$$\lambda^*(x, y, t | H_t) = \mu(x, y) + \sum_{t_i < t} g(x - x_i, y - y_i, t - t_i),$$

— Ces processus peuvent être marqués, dans ce cas g dépendra des marques.

Plusieurs questions se posent concernant le choix de la fonction d'excitation : la forme des termes spatiaux, les régions des événements et leur dépendance de la magnitude. Musmeci et Vere-Jones (1992) ont choisit comme fonction d'excitation la fonction :

$$g(x, y, t, m) = \frac{C \exp(\alpha m) \exp(-\beta t)}{2\pi\sigma_x\sigma_y t} \exp \left\{ \frac{-1}{2t} \left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} \right) \right\},$$

m est la magnitude, et x et y sont les coordonnées des locations. Les paramètres du modèle sont $(\alpha, \beta, \sigma_x^2, \sigma_y^2)$.

Extension du modèle ETAS

On s'intéresse dans la suite à l'extension du modèle ETAS à la dimension spatiale. Dans ce cas, on se restreint à des fonctions d'excitation telles que :

$$\lambda(t) = \iint_X \lambda(x, y, t) dx dy \quad (29)$$

avec λ l'intensité conditionnelle du modèle ETAS définie dans l'équation (26), ainsi

$$\iint_X g(x, y, t, m) \approx \iint_{\mathbb{R}^2} g(x, y, t, m) dx dy = \frac{K_0 \exp(\alpha(m - M_0))}{(t + c)^p}.$$

Parmi les fonctions vérifiant cette contrainte :

$$g(x, y, t, m) = \frac{K_0}{(t + c)^p} \exp \left\{ -\frac{1}{2} \frac{x^2 + y^2}{d \exp(\alpha(m - M_0))} \right\}. \quad (30)$$

les paramètres de ce modèle sont (K_0, c, α, p, d) .

En général

$$g(x, y, t, m) = k(m) \times \frac{(p-1)c^{(p-1)}}{(t+c)^{(p-1)}} \times \left[\frac{1}{\pi\sigma(m)} f\left\{ \frac{x^2 + y^2}{\sigma(m)} \right\} \right],$$

tel que $k(m)$ donne le nombre de descendants en fonction de la magnitude. $\frac{(p-1)c^{(p-1)}}{(t+c)^{(p-1)}}$ est la densité de probabilité temporelle et $\frac{1}{\pi\sigma(m)} f\left\{ \frac{x^2 + y^2}{\sigma(m)} \right\}$ représente la densité de probabilité spatiale.

Pour l'expression (30), on a $f(x) = \exp(-x)$ et $k(m) \propto \exp(-\alpha m)$. En effet, cette équation équivaut à :

$$g(x, y, t, m) = \exp(\alpha m) \frac{2\pi K_0 \exp(-\alpha M_0)}{(p-1)c^{(p-1)}} \times \frac{(p-1)c^{(p-1)}}{(t+c)^{(p-1)}} \\ \times \left[\frac{1}{\pi 2d \exp(\alpha(m-M_0))} f\left\{ \frac{x^2 + y^2}{2d \exp(\alpha(m-M_0))} \right\} \right].$$

14.2 Simulation

Dans cette section on s'intéresse à la simulation d'un processus de Hawkes spatio-temporel ayant la fonction (30) comme fonction d'excitation. D'après Ogata [11], on peut simuler ce type de processus en adaptant l'algorithme d'amincissement d'Ogata (1981) du cas unidimensionnel. Les points sont simulés en séquence, on simule la coordonnée temporelle par amincissement en utilisant l'intensité globale du modèle ETAS temporel, puis on simule les coordonnées spatiales et la marque selon leurs densités. On reprend ci-dessous le code d'Ogata [11]. On note λ_a l'intégration de l'intensité du modèle sur l'espace, qui est égale à l'intensité du modèle ETAS temporel d'après (29). On pose

$$\lambda_0 = \iint_X \mu(x, y) dx dy$$

et

$$\lambda_j(t) = \iint_{\mathbb{R}^2} g(x, y, t - t_j, M_j) dx dy$$

donc

$$\lambda_a(t) = \lambda_0 + \sum_{j=0}^J \lambda_j(t).$$

Étapes de la simulation

1. On simule le premier point t_1 du processus selon un Poisson homogène d'intensité λ_0 . On l'accepte si $t_1 < T$, $\lambda_{sup} = \lambda_0$.
2. On simule les coordonnées spatiales de ce point par amincissement selon un processus de Poisson non homogène sur l'espace d'intensité $\mu(x, y)$.
3. On simule la magnitude M et on pose $\lambda_{sup} = \lambda_a(t_1)$
4. $i = 2$, on simule un nouveau point candidat $s_i = s_{i-1} + w$ tel que w suit une loi exponentielle de paramètre λ_{sup} .
5. Si $s_i > T$ on arrête l'algorithme. Sinon, on simule une variable u uniforme entre 0 et 1.
6. Si $u < \frac{\lambda_a(s)}{\lambda_{sup}}$, on met $t_i = s_i$. Sinon, on pose $\lambda_{sup} = \lambda_a(s)$ et on revient à l'étape 4.
7. On simule une variable u uniforme entre 0 et 1, on choisit J le plus petit tel que $\sum_{j=0}^J \lambda_j(t) > u \lambda_a(t)$.

8. Si $J = 0$, alors on simule (x_i, y_i) sur X selon un processus de Poisson non homogène d'intensité $\mu(x, y)$ et on passe à l'étape 11.
9. Sinon si $J > 0$, on simule a, b qui suivent la densité de probabilité spatiale incluse dans l'équation (30) pour la magnitude m_J , ceci revient à simuler une loi normale centrée d'écart type $\sigma = \sqrt{d} \exp(\frac{\alpha(m - M_0)}{2})$ par la méthode de Box-Muller. On pose $x = x_J + a$ et $y = y_J + b$.
10. Si (x, y) n'est pas dans X on revient à l'étape 4.
11. On simule la magnitude m_i , et on accepte le point t_i avec ses localisations (x_i, y_i) . On pose $i = i + 1$ on revient à l'étape 4.

Le programme ci-dessous donne le code R pour simuler un processus de Hawkes spatio-temporel selon l'algorithme expliqué ci-dessus. On définit la région spatiale comme un rectangle délimité par des valeurs $x_{min}, x_{max}, y_{min}$ et y_{max} . Cet algorithme simule un processus de Hawkes avec une intensité de base $\mu(x, y)$ constante égale à une valeur λ . En effet, dans ce type de processus, μ est non-paramétrique, elle est estimée à partir des données.

```

params=list(k=0.05,
            c=0.5,
            p=1.2,
            lambda=1.2,
            d=0.06,
            T=100,
            alpha=1.2,
            beta=1.8,
            M0=5,
            xmin=-10^3,
            xmax=10^3,
            ymin=-10^3,
            ymax=10^3,
            model="Spatio-temporal")

# Simulation des magnitudes par inversion de la fonction de rpartition
#
# @param params Nombre de paramtres
# @param n Nombre de tirages
#
# @return n realisation de la magnitude
sim.magnitude=function(params,n){
  return(params$M0+rexp(n)/params$beta)
}
# simule n echantillon d'un processus de Poisson de paramtre a
Poisson.space=function(lambda,xmin,xmax,ymin,ymax){
  #lambda est l' intensit du processus
  #calcul de l' air du rectangle

```

```

x__dist=abs(xmax-xmin);y__dist=abs(ymax-ymin); #dimensions du
rectangle
air=x__dist*y__dist

#Simulation du processus
nbpoints=rpois(1,air*lambda);#nombre de points selon la loi de Poisson
x=x__dist*runif(nbpoints)+xmin;
y=y__dist*runif(nbpoints)+ymin;#y coordinates of Poisson points
loc=data.frame("x"=x,"y"=y)
return(loc)
}

#Simulation du processus
#
# @param param List of parameters
sim.Hawkes.space=function(params){
  t=c() # t est le vecteur des temps d'occurrence du processus
  m=c() # m est le vecteur des magnitudes
  loc=data.frame("x"=c(),"y"=c()) #tableau des localisations
  s=0 #temps d'occurrence des vnements
  # m et t sont des vecteurs
  # s est un scalaire
  loc__poisson=Poisson.space(params$lambda,params$xmin,params$xmax,params$ymin,params$ymax)
  #localisations d'un processus de poisson homogene de paramtres lambda
  intensity . local <-function(s,t,m,params)
  {
    return(params$lambda+sum(params$k*exp(params$alpha*(m-params$M0))/(params$c+s-t)^params$
  )
  }
  #pour trouver l' antcedent potentiel de l' vnement venir
  find . J<-function(s,u,t,m,params){
    somme=0;i=1
    while (u*intensity . local (s,t,m,params)>somme & i<=length(t)){
      mu=params$k*exp(params$alpha*(m[i]-params$M0))/(params$c+s-t[i])^params$p
      somme=somme+mu
      i=i+1
    }
    if (i<=length(t)){
      return(i-1)}else{return(0)}
  }
  lambda.sup=intensity.local(s,t,m,params) # intensit d'amincissement
  ##
  #simulation du premier point selon le processus de Poisson de base
  i=0 #aucun immigrant
  s=rexp(1,params$lambda)
  while (s>params$T) {
    s=rexp(1,params$lambda)
  }
}

```

```

}
t=c(t,s)          # actualisation du vecteur des dates d'occurrence
i=i+1 #nouveau immigrant
xy=c(loc_poisson$x[i],loc_poisson$y[i])# gnration des localisations
loc=rbind(loc,xy)
names(loc)[1] <- "x"
names(loc)[2] <- "y"
mag=sim.magnitude(params,1) #gnration de la magnitude
m=c(m,mag)
##
while (s<params$T) {
  s=s+rexp(1,lambda.sup) # incrmnte le prochain temps, exponentiel
  lambda.temp=intensity.local(s,t,m,params) # nouvelle intensit
  if (runif(1,0,1)<lambda.temp/lambda.sup){ # on accepte s avec la
    proba lambda.temp/lambda.sup
    u=runif(1,0,1)
    J=find.J(s,u,t,m,params)
    if (J==0){
      t=c(t,s)          #actualisation du vecteur des dates d'occurrence
      i=i+1 #nouveau immigrant
      xy=c(loc_poisson$x[i],loc_poisson$y[i])# gnration des
        localisations
      loc=rbind(loc,xy)
      mag=sim.magnitude(params,1) #gnration de la magnitude
      m=c(m,mag)
    }else{
      #extraire mj et (xj,yj)
      mj=m[J];xj=loc$x[J];yj=loc$y[J]
      #simulation de variables normales par Box Muller
      #rayon de Box Muller
      rayon<-function(u,mj,params){
        return(sqrt(params$d)*exp(-params$alpha*(mj-params$M0)/2)*sqrt(-2*log(u)))
      }
      u=runif(1,0,1);v=runif(1,0,1)
      r=rayon(u,mj,params)
      x=xj+r*cos(2*pi*v);y=yj+r*sin(2*pi*v)
      #on vrifie si les loc sont dans le domaine
      if (x<params$xmax & x>params$xmin & y<params$ymax &
        y>params$ymin){
        t=c(t,s)
        loc=rbind(loc,c(x,y))
        mag=sim.magnitude(params,1) #gnration de la magnitude
        m=c(m,mag)
      }
    }
  }
}
}

```

```

    lambda.sup=lambda.temp #on modifie lambda.sup chaque itration
  }
  idx<- t<=params$T
  t<- t[idx]
  m<- m[idx]
  loc=data.frame("x"=loc$x[idx],"y"=loc$y[idx])
  result <-
    data.frame(nbpnts=1:length(t),time=t,magnitudes=m,locations=loc)
  return( invisible ( result ))
}

res=sim.Hawkes.space(params)

```

14.3 Inférence

La log-vraisemblance d'un processus de Hawkes spatio-temporel s'écrit :

$$\ell = \sum_{i=1}^k \log(\lambda^*(t_i, x_i, y_i)) - \int_0^T \iint_X \lambda^*(t, x, y) dt dx dy. \quad (31)$$

Le premier terme de cette fonction se calcule directement et c'est le deuxième terme qui doit être simplifié. On suivra la démarche d'Ogata [9].

Pour simplifier, on suppose que l'intensité de base $\mu(x, y)$ est constante et que X est convexe. On définit l'ensemble $X - (a, b) = (x - a, y - b), (x, y) \in X$ pour chaque localisation (a, b) dans X . Puis, pour chaque événement i de localisation (x_i, y_i) , on suppose que la région $X - (x_i, y_i)$ peut être divisée radialement en K sous-régions $\{X_k^{(i)}, k = 1, 2, \dots, K\}$ par les segments radiaux reliant l'origine aux K noeuds à la frontière de la région convexe $X - (a, b)$ tels que

$$X - (x_i, y_i) = \cup_{k=1}^K X_k^{(i)}.$$

Soient θ_k et θ_{k+1} les angles des segments qui délimitent la région $X_k^{(i)}$ à partir de l'axe des abscisses dans le sens antihoraire. On note $\Delta k = \theta_{k+1} - \theta_k$. Ainsi, sachant que

$$\lambda^*(t, x, y) = \mu + \int_0^T \iint_X g(t-s, x-a, y-b) N(ds, da, db),$$

et en changeant l'ordre des intégrales, le second terme de la log-vraisemblance s'écrit :

$$\begin{aligned} \int_0^T \iint_X \lambda^*(t, x, y) dt dx dy &= \mu T |X| + \int_0^T \iint_X \int_{M_0}^\infty N(ds, da, db) \\ &\quad \times \int_0^T \iint_X g(t-s, x-a, y-b, M) dt dx dy, \end{aligned}$$

on fait un changement de variable

$$\begin{aligned} \int_0^T \iint_X \lambda^*(t, x, y) dt dx dy &= \mu T |X| + \int_0^T \iint_X \int_{M_0}^\infty N(ds, da, db) \\ &\quad \times \int_0^{T-s} \iint_{X-(a,b)} g(t, x, y, M) dt dx dy, \end{aligned}$$

donc

$$\begin{aligned} \int_0^T \iint_X \lambda^*(t, x, y) dt dx dy &= \mu T |X| + \sum_{i=1}^n \int_0^{T-t_i} \sum_{k=1}^K \iint_{X_k^{(i)}} g(t, x, y, M) dt dx dy \\ &\approx \mu T |X| + \sum_{i=1}^n \int_0^{T-t_i} \frac{K_0}{(t+c)^p} \sum_{k=1}^K S_k(x_i, y_i, M_i) \cdot \frac{\Delta k}{2\pi}, \end{aligned}$$

dans cette dernière étape, on approxime l'intégrale de g dans les coordonnées cartésiennes sur la région $X_k^{(i)}$ en une intégrale par rapport aux coordonnées polaires $rdrd\theta$ sur le disque centré sur l'origine et de rayon r_k donnant la distance entre (x_i, y_i) et un point dans le segment à la frontière de $X_k^{(i)}$. S_k est l'intégration du facteur spatial de $g(t, x, y, M)$ sur le disque de rayon r_k .

$$S_k(x_i, y_i, M_i) = 2\pi d \exp(\alpha(M_i - M_0)) \left(1 - \exp \left[\frac{-r_k^2}{2d \exp(\alpha(M_i - M_0))} \right] \right)$$

On maximise la log-vraisemblance en utilisant la méthode de Davidon-Fletcher-Powell.

15 Processus de Hawkes inhomogènes

15.1 Modèle ETAS non stationnaire

Kumazawa et Ogata (2016) [7] se sont intéressés à l'étude d'un modèle ETAS non stationnaire en choisissant des paramètres μ et k dans l'équation (26) qui dépendent du temps, afin d'examiner les effets induits par des facteurs exogènes. k représente le facteur de production de descendants par rapport à l'amplitude de la magnitude, il est proportionnel au nombre de séismes déclenchés d'une magnitude supérieure à la magnitude seuil. Le taux de sismicité de fond résume le taux d'occurrence dû aux effets à long terme des grands tremblements de terre passés qui se sont produits avant la cible période d'analyse ainsi que le taux d'occurrence dû des événements en dehors de la région de concentration. Ces deux paramètres k et μ sont les plus sensibles à la sismicité, en plus il y a différents effets sont absorbés dans μ dans le cadre stationnaire du modèle 'ETAS et sont souvent la source d'estimations biaisées. Dans le cas du modèle non stationnaire ces effets seront inclus dans un taux μ dépendant du temps.

On introduit des fonctions de correction qui sont estimées par une méthode bayésienne avec fixation des paramètres de référence du modèle ETAS stationnaire :

$$\mu(t) = \mu q_\mu(t), k(t) = k q_k(t).$$

16 Conclusion

Dans ce travail, nous avons fourni une modélisation mathématique des processus de Hawkes, selon 3 modèles : le modèle exponentiel, le modèle puissance et le modèle ETAS. Nous avons présenté une méthode de simulation de ces processus, basée sur l'algorithme d'amincissement qui a été implémentée sur R. Et nous avons développé des procédures d'inférence pour estimer les paramètres des modèles par maximisation de la vraisemblance et les valider à travers des tests statistiques.

Les résultats obtenus sur des données sismiques ont montré que les processus de Hawkes sont effectivement un outil mathématique intéressant pour ce type de données. L'intégration des magnitudes dans le modèle marqué ETAS a permis d'améliorer les résultats par rapport aux modèles exponentiel et puissance. Ces résultats peuvent être encore améliorés en ajoutant la dimension spatiale des données pour passer aux processus de Hawkes spatio-temporels.

Références

- [1] S. Asmussen. *Applied Probability and Queues*. Applications of Mathematics : Stochastic Modelling and Applied Probability. 2 edition, 2003.
- [2] Yuanda Chen. Thinning algorithms for simulating point processes, 2016.
- [3] Daryl J Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes, Volume I : Elementary Theory and Methods*. Springer Science & Business Media, 2 edition, 2003.
- [4] Roger Fletcher and Michael JD Powell. A rapidly convergent descent method for minimization. *The computer journal*, 6(2) :163–168, 1963.
- [5] Beno Gutenberg and Charles F Richter. Frequency of earthquakes in california. *Bulletin of the Seismological Society of America*, 34(4) :185–188, 1944.
- [6] Ivan L Johnson Jr. The davidon-fletcher-powell penalty function method : a generalized iterative technique for solving parameter optimization problems. 1976.
- [7] Takao Kumazawa and Yosihiko Ogata. Quantitative description of induced seismic activity before and after the 2011 tohoku-oki earthquake by nonstationary etas models. *Journal of Geophysical Research : Solid Earth*, 118(12) :6165–6182, 2013.
- [8] Patrick J Laub, Thomas Taimre, and Philip K Pollett. Hawkes processes. *arXiv preprint arXiv :1507.02822*, 2015.
- [9] Y Ogata. Statistical model for standard seismicity and detection of anomalies by residual analysis. *Tectonophysics*, 169(1-3) :159–174, 1989.
- [10] Yosihiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical association*, 83(401) :9–27, 1988.
- [11] Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2) :379–402, 1998.
- [12] Yosihiko Ogata. Exploratory analysis of earthquake clusters by likelihood-based trigger models. *Journal of Applied Probability*, 38(A) :202–212, 2001.
- [13] Yosihiko Ogata, Ritsuko S Matsu’ura, and Koichi Katsura. Fast likelihood computation of epidemic type aftershock-sequence model. *Geophysical research letters*, 20(19) :2143–2146, 1993.
- [14] Tohru Ozaki. Maximum likelihood estimation of hawkes’ self-exciting point processes. *Annals of the Institute of Statistical Mathematics*, 31(1) :145–155, 1979.
- [15] Tokuji Utsu, Yosihiko Ogata, et al. The centenary of the omori formula for a decay law of aftershock activity. *Journal of Physics of the Earth*, 43(1) :1–33, 1995.
- [16] Margaret H Wright. Interior methods for constrained optimization. *Acta numerica*, 1 :341–407, 1992.

- [17] Jiancang Zhuang, Yoshihiko Ogata, and David Vere-Jones. Stochastic de-clustering of space-time earthquake occurrences. *Journal of the American Statistical Association*, 97(458) :369–380, 2002.