

ÉCOLE NORMALE SUPÉRIEURE DE RENNES  
Département de Mathématiques

# Quantifier la synchronisation entre neurones

Magali JAY

Stage de fin de licence

Encadré par

Laurent BOUGRAIN, INRIA-Loria, Nancy  
et Radu RANTA, CRAN, Nancy

23 Mai - 13 Juillet 2018

## REMERCIEMENTS

Je tiens à remercier Laurent Bougrain et Radu Ranta qui ont co-encadré mon stage et m'ont accueillie dans leurs équipes, ainsi qu'Anne Gégout-Petit, de l'institut de mathématiques Élie Cartan de Nancy, qui m'a expliqué le principe du test par permutation.

Merci aussi à tous les membres des deux équipes, permanents, stagiaires et thésards avec qui j'ai aimé discuter et partager la vie des laboratoires.

Et merci aux relecteurs pour leurs remarques précieuses !

### Résumé

Après une brève présentation du fonctionnement neuronal, on expose deux façons de mesurer la synchronisation entre neurones. Afin de différencier des neurones sans rapport et des neurones corrélés, on effectue un test statistique. Pour cela, on doit choisir un modèle. Dans le cadre de ce rapport, on étudie le modèle du processus de Poisson. Cela permet de mener les tests statistiques et de conclure ou non à une synchronisation.

**Mots-clés :** trains de spikes, synchronisation, probabilité, processus de Poisson, tests statistiques

### Abstract

This report begins with a short explanation of how neurons work. Then distances of synchrony between spike trains are presented. Statistical tests are used in order to distinguish correlated spike trains from independent ones. One needs to suppose that neurons' activity follows a probabilistic law. This report studies the Poisson point process.

**Key words :** spike trains, synchrony, probability, Poisson point process, statistical tests

# Table des matières

- 1 Introduction . . . . . 3
- 2 Distances de synchronie . . . . . 4
  - 2.1 Distance de Victor et Purpura . . . . . 4
  - 2.2 Distance de van Rossum . . . . . 6
  - 2.3 Distances sans paramètre : ISI et SPIKE . . . . . 8
  - 2.4 Signification d'une distance . . . . . 8
- 3 Cadre théorique : processus de Poisson . . . . . 9
  - 3.1 Définitions . . . . . 9
  - 3.2 Vecteurs à densité . . . . . 10
  - 3.3 Temps d'attente . . . . . 10
  - 3.4 Une autre construction . . . . . 13
  - 3.5 Généralisation : processus non stationnaire . . . . . 17
  - 3.6 Modélisation . . . . . 17
- 4 Analyse statistique . . . . . 20
  - 4.1 Tests statistiques . . . . . 21
  - 4.2 Tester la synchronisation . . . . . 21
  - 4.3 Conclusion . . . . . 31
  
- Bibliographie . . . . . 32**

# 1 Introduction

Commençons par un bref aperçu du contexte biologique de l'étude. Les précisions données ci-dessous permettent de comprendre les modèles choisis et les questions qui se posent. J'ai également ajouté quelques détails qui me semblaient intéressants, et qui n'ont d'autre but que d'éveiller la curiosité du lecteur.

Le cerveau compte environ cent milliards de neurones, interagissant *via* des connections synaptiques. Chaque neurone peut produire une impulsion électrique (créée par des accumulations d'ions dans la cellule, essentiellement sodium et potassium). On appelle cette impulsion *potentiel d'action*, ou *spike*. Ce potentiel d'action est transmis au niveau de *synapses* à d'autres neurones, appelés *post-synaptiques*. Chaque neurone peut recevoir des spikes de plusieurs neurones. En fonction des spikes reçus, il produira ou non à son tour une impulsion. Ainsi, le neurone intègre de nombreuses informations, qui lui permettent de déclencher à son tour une impulsion, ou non.

Les potentiels d'action se propagent le long des axones à une vitesse allant jusqu'à une centaine de mètres par seconde. Autour de la plupart des axones, une couche de myéline facilite cette propagation.

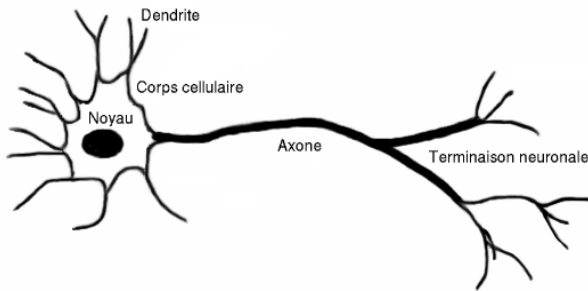


FIGURE 1 – Neurone [1]

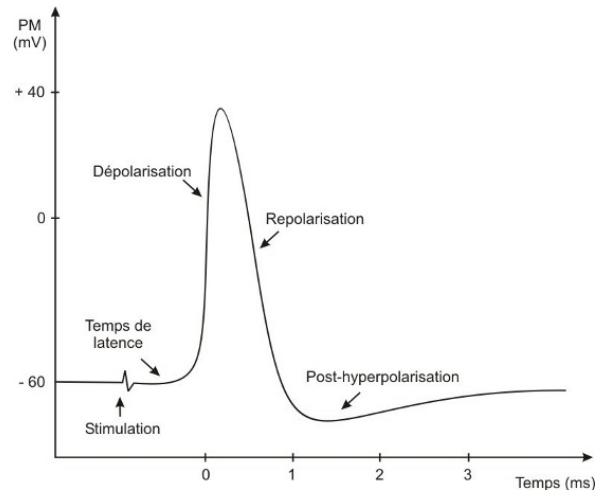


FIGURE 2 – Potentiel d'action [2]

Compte tenu de la forme d'un potentiel d'action – un pic d'une à deux millisecondes –, on choisit souvent de représenter les *trains de spikes* par des séquences binaires. Si on enregistre l'activité d'un neurone pendant une seconde avec une résolution temporelle d'une milliseconde, le train de spike correspondant est une suite de longueur 1000, où le  $i$ -ème terme représente l'activité à la  $i$ -ème milliseconde, 0 s'il n'y a pas eu de spike, 1 s'il y en a eu.

Même si chaque potentiel d'action a une forme similaire (cf. FIGURE 2), tous les potentiels d'action n'ont pas exactement la même forme. Cela permet de distinguer un spike émis par un neurone d'un spike émis par un autre neurone.

En pratique, on peut mesurer l'activité des neurones :

- à l'échelle macroscopique, avec des électrodes posées sur la tête du sujet, c'est *l'électroencéphalographie*, qui mesure la somme des activités de millions de neurones ;
- à l'échelle microscopique, avec des électrodes d'une vingtaine de micromètres, implantées dans le cerveau du sujet à l'aide d'une tige de quelques millimètres de diamètre, on mesure alors la somme des activités de quelques neurones, typiquement moins d'une dizaine.

Les neuroscientifiques se sont d'abord intéressés à la fréquence (et ses variations) des potentiels d'action, en supposant qu'elle constituait le moyen pour les neurones de coder l'information. Depuis, des études ont montré que plusieurs neurones qui envoient un spike relativement en même temps vers un même autre neurone sont plus efficaces que lorsqu'ils ne sont pas coordonnés : il faut beaucoup plus de sollicitations sur un neurone pour qu'il émette un spike à son tour si ces sollicitations ne se produisent pas dans un laps de temps court ([7]). Cependant, je n'ai trouvé aucune précision sur la valeur de ce laps de temps.

De plus, certaines fonctions cérébrales, comme la commande d'un mouvement, s'accompagnent de synchronisation ou dé-synchronisation des neurones.

Ces observations motivent l'étude de la synchronie entre trains de spikes. Les chercheurs débattent encore aujourd'hui du rôle et de l'origine de la synchronie dans le codage neuronal de l'information. Face à ce qui semble être deux trains de spikes synchrones, il est donc essentiel de savoir si la synchronie est significative ou si elle peut être simplement due au hasard.

Nous nous intéressons à la synchronisation entre deux trains de spikes. Différentes mesures ont déjà été proposées par plusieurs chercheurs ([14], [12], [11], [8]). Elles quantifient la synchronisation entre deux neurones. Il s'agit alors de savoir ce que signifie la valeur obtenue. Est-ce que la synchronisation est due simplement au hasard (les deux phénomènes sont indépendants) ou traduit-elle une réelle corrélation entre les deux phénomènes ?

## 2 Distances de synchronie

Cette partie présente plusieurs façons de mesurer la synchronie entre deux trains de spikes, tirées de [14]. De nombreuses autres mesures sont utilisées ([14], [12], [11], [8]).

### 2.1 Distance de Victor et Purpura

**Définition 1** (transformation d'un train de spikes). *Soient deux trains de spikes  $X$  et  $Y$ . On appelle transformation de  $X$  à  $Y$  une suite de suppression(s), ajout(s) et/ou décalage(s) où :*

- *supprimer un spike de  $X$ , c'est remplacer un 1 par un 0 dans  $X$ .*
- *ajouter un spike à  $X$ , c'est remplacer un 0 par un 1 dans  $X$ .*
- *décaler un spike de  $X$  vers la gauche (resp. vers la droite), c'est remplacer un motif 01 par 10 (resp. 10 par 01).*

Une transformation de  $X$  à  $Y$  définit naturellement une transformation de  $Y$  à  $X$ , il suffit de *l'inverser*, c'est-à-dire de la parcourir à partir de la fin et d'échanger suppression et ajout, décalage à droite et à gauche.

Par exemple, on peut passer de  $X = 00101001$  à  $Y = 11001001$  par les transformations suivantes :

transformation A	transformation B	transformation C
00101001	00101001	00101001
01001001	01001001	0001001
10001001	10001001	01001001
10010001	11001001	11001001
10100001		
11000001		
11001001		

On a représenté en **bleu** les décalages et en **rouge** les ajouts et suppressions.

On peut bien sûr imaginer une infinité de transformations pour passer de  $X$  à  $Y$ , on se limitera ici à ces trois exemples.

**Remarque 1.** Ces trains de spikes sont choisis pour donner un exemple intéressant, mais si on considère que chaque bit représente une milliseconde, alors  $X$  et  $Y$  ne sont pas réalistes d'un point de vue biologique. En effet, la fréquence d'émission de spikes varie de 4 à 80 Hz, et l'émission d'un potentiel d'action est suivi d'une période, dite *réfractaire*, pendant laquelle le neurone n'émet pas de spike. Pour simplifier, nous considérerons quand même que chaque bit correspond à une milliseconde.

**Définition 2** ( $q$ -longueur d'une transformation). Soient deux trains de spikes  $X$  et  $Y$ . Soit  $q \in \mathbb{R}_+^*$ . Soit  $t = (t_k)_{1 \leq k \leq N}$  une transformation de  $X$  à  $Y$ , comprenant  $n_s$  suppressions,  $n_a$  ajouts et  $n_d$  décalages. On appelle  $q$ -longueur de  $t$  la quantité  $n_s + n_a + qn_d$ .

Par exemple, les transformations A, B et C du paragraphe précédent ont pour  $q$ -longueurs :

$q$	transformation A	transformation B	transformation C
0	1	1	3
0.5	3.5	2	3
1	6	3	3
1.5	8.5	4	3

Selon le paramètre  $q$ , la transformation de  $q$ -longueur minimale n'est pas la même. Quand  $q$  est faible, on a intérêt à privilégier les décalages aux suppressions et ajouts. Quand  $q$  est élevé, il vaut mieux (pour minimiser la  $q$ -longueur) utiliser des suppressions et ajouts que des décalages.

**Définition 3.** Soient deux trains de spikes  $X$  et  $Y$ . Soit  $q \in \mathbb{R}_+^*$ .

La distance de Victor et Purpura de coût  $q$  est le minimum des  $q$ -longueurs des transformations de  $X$  à  $Y$ . On la note  $d_{VP}(X, Y)$ .

*Démonstration.* L'ensemble des transformations de  $X$  à  $Y$  est non vide puisqu'il contient la transformation qui consiste à supprimer tous les spikes de  $X$  et à lui ajouter tous les spikes de  $Y$ . De plus la  $q$ -longueur d'une transformation est positive. L'ensemble des  $q$ -longueurs des transformations de  $X$  à  $Y$  admet donc une borne inférieure. Enfin, on peut se limiter aux transformations qui ne font pas d'actions contraires, comme supprimer puis ajouter le même spike, ou alors transformer un motif 01 en 10 puis le transformer à nouveau en 01. Comme les trains de spikes sont finis, il y a un nombre fini de telles transformations. La borne inférieure est donc un minimum.

On définit bien une distance :

- Elle est positive.
- Elle est symétrique puisque chaque transformation de  $X$  à  $Y$  définit une transformation de  $Y$  à  $X$ , de même  $q$ -longueur.
- Elle est nulle si et seulement si on peut transformer  $X$  en  $Y$  sans suppression, ni ajout, ni décalage (car  $q > 0$ ), c'est-à-dire si  $X$  et  $Y$  sont identiques.
- Elle vérifie l'inégalité triangulaire : si  $t_1$  et  $t_2$  transforment respectivement  $X$  en  $Y$  et  $Y$  en  $Z$  avec une  $q$ -longueur minimale, alors la concaténation de  $t_1$  et  $t_2$  transforme  $X$  en  $Z$ , et sa  $q$ -longueur est la somme des  $q$ -longueurs de  $t_1$  et  $t_2$ . Donc  $d_{VP}(X, Z) \leq d_{VP}(X, Y) + d_{VP}(Y, Z)$ .

□

D'après l'observation précédente, la distance de Victor et Purpura a deux comportements distincts. Quand  $q$  tend vers 0, on mesure la différence de nombre de spikes entre  $X$  et  $Y$ . Quand  $q$  tend vers  $+\infty$ , on mesure le nombre de spikes non coïncidents entre  $X$  et  $Y$ . Toujours avec les mêmes trains de spikes, on obtient, en fonction de  $q$  :

$q$	$d_{VP}(X, Y)$
0	1
0.5	1.5
1	2
1.5	2.5

### Programmation efficace

Quand on travaille sur des populations de  $N$  neurones, on peut vouloir calculer la synchronisation de cette population. Une solution est de calculer les distances (par exemple de Victor et Purpura) entre tous les neurones, puis de faire la moyenne. Cela demande donc  $\frac{N(N-1)}{2}$  calculs de distance. Pour des populations grandes, il est très important d'avoir un programme efficace pour le calcul de distance.

J'ai utilisé un programme, proposé par [3], qui s'effectue en  $O(nm)$  où  $n$  désigne le nombre de spikes de  $X$  et  $m$  celui de  $Y$ . Pour une population de 200 neurones, émettant des spikes avec une moyenne de 10 Hz pendant 1 seconde, le calcul prend 3 secondes. S'ils émettent pendant 3 secondes, le calcul prend cette fois 20 secondes.

Pour appliquer cette mesure sur de plus grandes populations (il y a par exemple environ  $10^7$  neurones dans l'hippocampe), il faut trouver d'autres solutions. On pourrait éventuellement utiliser plusieurs processeurs en parallèle, ou encore se contenter d'effectuer les calculs sur un échantillon de données. Dans la suite, nous ne nous posons pas ces questions, et nous concentrerons sur des populations de tailles raisonnables.

## 2.2 Distance de van Rossum

La distance de van Rossum est une autre distance dépendant d'un paramètre.

À chaque train de spike  $X = (x_i)_{1 \leq i \leq n}$ , on associe la fonction continue par morceaux  $f_X$  telle que

$$\forall t \in \mathbb{R}, f_X(t) = \sum_{k=1}^n x_k h_k(t)$$

où  $h_k$  est définie par :

$$\forall t \in \mathbb{R}, h_k(t) = \exp\left(-\frac{t - ks}{\tau_R}\right) \mathbb{1}_{t > ks}$$

où  $s$  désigne la précision temporelle, c'est-à-dire la durée que représente chaque bit de  $X$ , et  $\mathbb{1}_{t > ks}$  vaut 1 si  $t > ks$  et 0 sinon.

Par exemple, en reprenant l'exemple précédant,  $X = 00101001$  et  $Y = 11001001$ , avec  $\tau_R = 0.2$  ms, on obtient :

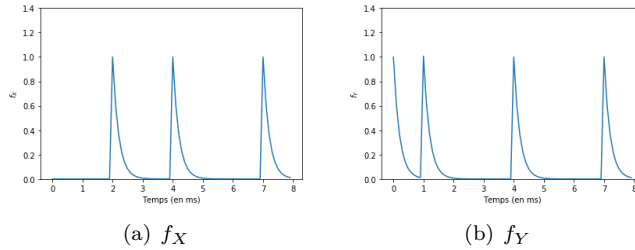


FIGURE 3 – fonctions associées à  $X$  et  $Y$  pour  $\tau_R = 0.2$  ms

Pour  $\tau_R = 1$  ms, on obtient :

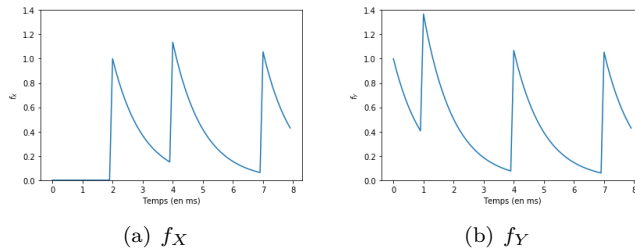


FIGURE 4 – fonctions associées à  $X$  et  $Y$  pour  $\tau_R = 1$  ms

**Définition 4** (distance de van Rossum). On définit la distance de van Rossum de paramètre  $\tau_R > 0$  par :

$$D_R(X, Y) = \sqrt{\frac{2}{\tau_R} \int_0^\infty |f_X(t) - f_Y(t)|^2 dt}$$

*Démonstration.* Cela définit bien une distance : à un facteur multiplicatif près, c'est la distance  $L^2$  entre  $f_X$  et  $f_Y$  (la séparation est assurée par le fait que  $Z \mapsto f_Z$  est injective). □

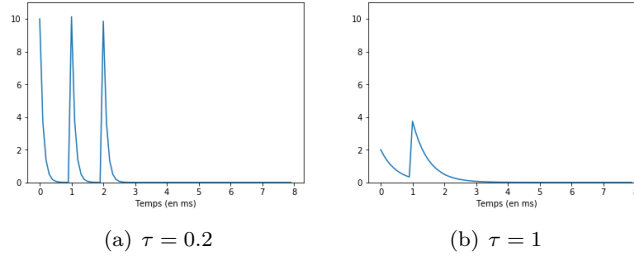


FIGURE 5 –  $\frac{2}{\tau_R}(f_X - f_Y)^2$  pour  $\tau = 0.2$  ms et  $\tau = 1$  ms

$\tau_R$  est le temps caractéristique qui définit ce qu'on appelle synchronie : plus il est grand, plus deux spikes éloignés temporellement pourront être considérés comme synchrones.

Voici différentes valeurs obtenues avec l'exemple précédent :

$\tau_R$	$D_R(X, Y)$
0.01	1.73
0.2	1.73
1	1.77
2	1.94

### Programmation efficace

J'ai programmé l'algorithme présenté dans l'article de Thomas KREUZ et Conor HOUGHTON [13], à partir de la version proposée sur le site de Thomas KREUZ [4].

Tout d'abord, la distance s'exprime avec une intégrale mais compte tenu de la forme de  $f_X$  et  $f_Y$  (somme d'exponentielles), on peut la calculer autrement (cf. calcul ci-dessous). Pour le calcul et pour la représentation en machine, on représente un train de spikes non plus comme une suite de 0 et de 1 mais comme la suite des instants de spikes, en nombre d'unité de temps (ici 1 ms). Avec l'exemple précédent, on a  $X = (2, 4, 7)$  et  $Y = (0, 1, 4, 7)$ .

On note  $X = (u_i)_{1 \leq i \leq N}$  et  $Y = (v_i)_{1 \leq i \leq M}$ , avec  $N$  le nombre de spikes de  $X$  et  $M$  le nombre de spikes de  $Y$ . Alors  $u_k$  (resp.  $v_k$ ) désigne l'instant du  $k$ -ième spike de  $X$  (resp. de  $Y$ ).

$$\begin{aligned}
D_R(X, Y)^2 &= \frac{2}{\tau_R} \int_0^\infty |f_X(t) - f_Y(t)|^2 dt \\
&= \frac{2}{\tau_R} \int_0^\infty \left( \sum_{i=1}^N \exp\left(-\frac{t-u_i}{\tau_R}\right) \mathbb{1}_{t \geq u_i} - \sum_{j=1}^M \exp\left(-\frac{t-v_j}{\tau_R}\right) \mathbb{1}_{t \geq v_j} \right)^2 dt \\
&= \frac{2}{\tau_R} \int_0^\infty \left[ \sum_{i=1}^N \sum_{j=1}^N \exp\left(-\frac{t-u_i}{\tau_R}\right) \exp\left(-\frac{t-u_j}{\tau_R}\right) \mathbb{1}_{t \geq u_i} \mathbb{1}_{t \geq u_j} \right. \\
&\quad \left. + \sum_{i=1}^M \sum_{j=1}^M \exp\left(-\frac{t-v_i}{\tau_R}\right) \exp\left(-\frac{t-v_j}{\tau_R}\right) \mathbb{1}_{t \geq v_i} \mathbb{1}_{t \geq v_j} \right. \\
&\quad \left. - 2 \sum_{i=1}^N \sum_{j=1}^M \exp\left(-\frac{t-u_i}{\tau_R}\right) \exp\left(-\frac{t-v_j}{\tau_R}\right) \mathbb{1}_{t \geq u_i} \mathbb{1}_{t \geq v_j} \right] dt \\
&= \frac{2}{\tau_R} \sum_{i=1}^N \sum_{j=1}^N \int_{\max(u_i, u_j)}^\infty \exp\left(-\frac{2t-u_i-u_j}{\tau_R}\right) dt \\
&\quad + \frac{2}{\tau_R} \sum_{i=1}^M \sum_{j=1}^M \int_{\max(v_i, v_j)}^\infty \exp\left(-\frac{2t-v_i-v_j}{\tau_R}\right) dt \\
&\quad - \frac{4}{\tau_R} \sum_{i=1}^N \sum_{j=1}^M \int_{\max(u_i, v_j)}^\infty \exp\left(-\frac{2t-u_i-v_j}{\tau_R}\right) dt \\
&= \frac{2}{\tau_R} \sum_{i=1}^N \sum_{j=1}^N \frac{\tau_R}{2} \exp\left(-\frac{2 \max(u_i, u_j) - u_i - u_j}{\tau_R}\right) \\
&\quad + \frac{2}{\tau_R} \sum_{i=1}^M \sum_{j=1}^M \frac{\tau_R}{2} \exp\left(-\frac{2 \max(v_i, v_j) - v_i - v_j}{\tau_R}\right) \\
&\quad - \frac{4}{\tau_R} \sum_{i=1}^N \sum_{j=1}^M \frac{\tau_R}{2} \exp\left(-\frac{2 \max(u_i, v_j) - u_i - v_j}{\tau_R}\right) \\
&= \sum_{i=1}^N \sum_{j=1}^N \exp\left(-\frac{|u_i - u_j|}{\tau_R}\right) + \sum_{i=1}^M \sum_{j=1}^M \exp\left(-\frac{|v_i - v_j|}{\tau_R}\right) - 2 \sum_{i=1}^N \sum_{j=1}^M \exp\left(-\frac{|u_i - v_j|}{\tau_R}\right)
\end{aligned}$$

Ensuite, on peut utiliser un tableau auxiliaire, appelé de marquage, qui évite de faire un calcul de somme double et permet de passer d'une complexité en  $O(MN)$ , où  $M$  et  $N$  désignent le nombre de spikes dans chacun des deux trains de spikes, à une complexité en  $O(M + N)$ .

### 2.3 Distances sans paramètre : ISI et SPIKE

D'autres distances sont définies. Par exemple, les distances ISI et SPIKE [14] ont l'intérêt de ne dépendre d'aucun paramètre. Le temps caractéristique du train de spikes est calculé localement, en fonction de la durée entre les spikes. SPIKE est une distance instantanée, elle est calculée à chaque instant, ce qui permet de détecter si la synchronisation est localisée dans le temps.

### 2.4 Signification d'une distance

Une fois que l'on a obtenu la distance entre deux trains de spikes, on a besoin d'une échelle pour savoir si cette valeur correspond à une synchronisation particulière entre les neurones ou si ce sont juste deux

neurones indépendants. Pour pouvoir établir cette échelle, il faut définir précisément le comportement auquel on s'attend de la part du neurone, c'est-à-dire le modèle probabiliste suivi par le neurone (ou en tout cas, une première approximation dans laquelle on se place).

### 3 Cadre théorique : processus de Poisson

Le modèle le plus simple à étudier est le processus de Poisson. Ce cadre traduit généralement assez mal le comportement réel d'un neurone, mais il a plusieurs intérêts : il constitue un cas de base assez simple, permettant une approche analytique ; il peut être adapté pour définir d'autres modèles, plus fidèles à la réalité ; enfin, d'un point de vue pédagogique, il permet de se familiariser avec les notions liées aux processus aléatoires.

Dans la suite, on se place dans  $(\Omega, \mathcal{F}, \mathbb{P})$  un espace probabilisé.

#### 3.1 Définitions

**Définition 5** (processus aléatoire).

Un processus aléatoire est une famille de variables aléatoires indexées par un ensemble  $T$ .

Le plus souvent  $T = \mathbb{R}, \mathbb{R}_+, \mathbb{Z}$  ou  $\mathbb{N}$  représente le temps. On dit alors que le processus est respectivement *continu* ou *discret*. Si  $T$  est fini, le processus est un vecteur aléatoire.

**Définition 6** (processus de comptage).

On appelle processus de comptage un processus aléatoire  $(N_t)_{t \in T}$  tel que :

- Les variables aléatoires  $(N_t)_{t \in T}$  sont à valeurs dans  $\mathbb{N}$ .
- $\forall \omega \in \Omega$ , la fonction  $t \in T \mapsto N_t(\omega)$  est croissante.

Un processus de comptage permet de modéliser un phénomène se répétant aléatoirement au fil du temps. Par exemple, des désintégrations radioactives, des accidents d'avions, les potentiels d'actions d'un neurone, *etc.*  $N_t$  représente le nombre de phénomènes survenus entre le temps 0 et le temps  $t$ .

Un processus de Poisson est un cas particulier de processus de comptage.

**Définition 7** (processus de Poisson). Soit  $\lambda \in \mathbb{R}_+^*$

Un processus de Poisson d'intensité  $\lambda$  est un processus de comptage  $(N_t)_{t \in \mathbb{R}_+}$  dont les accroissements sont stationnaires (1), indépendants (2) et suivent une loi de Poisson (3) :

- $\forall t \in \mathbb{R}_+, \forall s > 0, (N_{t+s} - N_t) \sim N_s$  (1)
- $\forall k \in \mathbb{N}, \forall t_1 < \dots < t_k, (N_{t_2} - N_{t_1}), \dots, (N_{t_k} - N_{t_{k-1}})$  sont indépendants (2)
- $\forall t \in \mathbb{R}_+, N_t \sim \mathcal{P}(\lambda t)$  (3)

**Remarque 2.** On applique la convention  $0^0 = 1$  pour définir la loi de Poisson de paramètre 0.  $N_0$  est alors  $\mathbb{P}$ -presque sûrement nulle.

Choisir de modéliser un phénomène aléatoire par un processus de Poisson, c'est postuler la loi que suivent les variables  $N_t$  et surtout que les accroissements sont indépendants, c'est-à-dire que la connaissance des événements passés (nombre de désintégrations radioactives jusqu'à présent, répartition de ces désintégrations au fil du temps) ne nous donne *aucune information* sur ce qui pourra se produire à l'avenir. En ce sens, un processus de Poisson traduit bien l'idée qu'on l'on se fait du hasard.

**Remarque 3.** Soit  $(N_t)_{t \in \mathbb{R}_+}$  un processus de Poisson.

Alors  $\mathbb{P}(N_t \geq 2) = 1 - \mathbb{P}(N_t = 1) - \mathbb{P}(N_t = 0) = 1 - \lambda t e^{-\lambda t} - e^{-\lambda t} = o(t)$   $\underset{t \rightarrow 0}{=}$ . La probabilité d'avoir deux événements survenant en même temps est négligeable. C'est cette propriété qui vaut à la loi de Poisson d'être appelée «loi des événements rares». Dans le cas qui nous intéresse ici, cela correspond au fait qu'un neurone n'émet qu'un potentiel d'action à la fois.

**Remarque 4.** On sait que l'espérance d'une variable aléatoire suivant une loi de Poisson est égale au paramètre de la loi. Dans notre contexte, cela veut dire que la fréquence moyenne d'émission d'un spike est donné par le paramètre du processus de Poisson, d'où le nom d'intensité.

### 3.2 Vecteurs à densité

Voici deux définitions relatives aux vecteurs aléatoires, dont nous aurons besoin par la suite.

**Définition 8** (fonction de répartition).

Soit  $(X_1, X_2, \dots, X_n)$  un vecteur aléatoire à valeurs dans  $\mathbb{R}^n$ .

La fonction de répartition de  $(X_1, X_2, \dots, X_n)$  est définie par :

$$\forall (x_1, \dots, x_n) \in \mathbb{R}^n, \quad F_{(X_1, \dots, X_n)}(x_1, \dots, x_n) = \mathbb{P}(X_1 \leq x_1 \cap \dots \cap X_n \leq x_n)$$

**Définition 9** (densité).

Soit  $(X_1, X_2, \dots, X_n)$  un vecteur aléatoire à valeurs dans  $\mathbb{R}^n$ .

On dit que  $(X_1, X_2, \dots, X_n)$  admet une densité lorsque pour tout borélien  $B \in \mathcal{B}(\mathbb{R}^n)$  de mesure de Lebesgue nulle,  $\mathbb{P}((X_1, \dots, X_n) \in B) = 0$ .

On peut aussi définir un vecteur à densité de la manière suivante :

**Définition 10** (densité).

Soit  $(X_1, X_2, \dots, X_n)$  un vecteur aléatoire à valeurs dans  $\mathbb{R}^n$ .

On dit que  $(X_1, X_2, \dots, X_n)$  admet une densité lorsqu'il existe  $f_{(X_1, \dots, X_n)} : \mathbb{R}^n \rightarrow \mathbb{R}$  une fonction réelle de  $n$  variables telle que :

$$\forall (x_1, \dots, x_n) \in \mathbb{R}^n, \quad F_{(X_1, \dots, X_n)}(x_1, \dots, x_n) = \int_{]-\infty, x_1] \times \dots \times ]-\infty, x_n]} f_{(X_1, \dots, X_n)}(t_1, \dots, t_n) dt_1 \dots dt_n.$$

**Remarque 5.** Comme les  $]-\infty, x_1] \times \dots \times ]-\infty, x_n]$  engendrent les boréliens de  $\mathbb{R}_+$ , et que  $F_{(X_1, \dots, X_n)}(x_1, \dots, x_n) = \mathbb{P}(X_1 \leq x_1 \cap \dots \cap X_n \leq x_n)$ , cette définition équivaut à définir  $f_{(X_1, \dots, X_n)}$  par :

$$\forall B \in \mathcal{B}(\mathbb{R}^n), \quad \mathbb{P}((X_1, \dots, X_n) \in B) = \int_B f_{(X_1, \dots, X_n)}(t_1, \dots, t_n) dt_1 \dots dt_n$$

Les définitions 9 et 10 sont bien sûr équivalentes, d'après le théorème de Radon-Nikodym :

**THÉORÈME 1** (Radon-Nikodym). Si  $\mu$  est une mesure absolument continue<sup>1</sup> par rapport à la mesure de Lebesgue  $\lambda_n$ , alors il existe une unique fonction  $h \in L^1(\mathbb{R}^n)$  telle que

$$\forall B \in \mathcal{B}(\mathbb{R}^n), \quad \mu(B) = \int_B h d\lambda_n = \int_{\mathbb{R}^n} h \mathbb{1}_B d\lambda_n.$$

On admet ce théorème. On pourra consulter [15] pour une démonstration.

### 3.3 Temps d'attente

Un processus de Poisson permet de compter le nombre d'événements qui surviennent pendant un laps de temps donné. On peut également s'intéresser au temps qui s'écoule entre deux phénomènes. On a vu que  $N_t$  représente le nombre de phénomènes qui se sont produits entre 0 et  $t$ . On considère maintenant la variable  $S_k$ , qui représente le temps écoulé jusqu'au phénomène numéro  $k$ , et  $X_k$  le temps écoulé entre les phénomènes numéro  $k - 1$  et numéro  $k$  (cf. FIGURE 6).

1. c'est-à-dire :  $\forall B \in \mathcal{B}(\mathbb{R}^n), \lambda_n(B) = 0 \Rightarrow \mu(B) = 0$ .

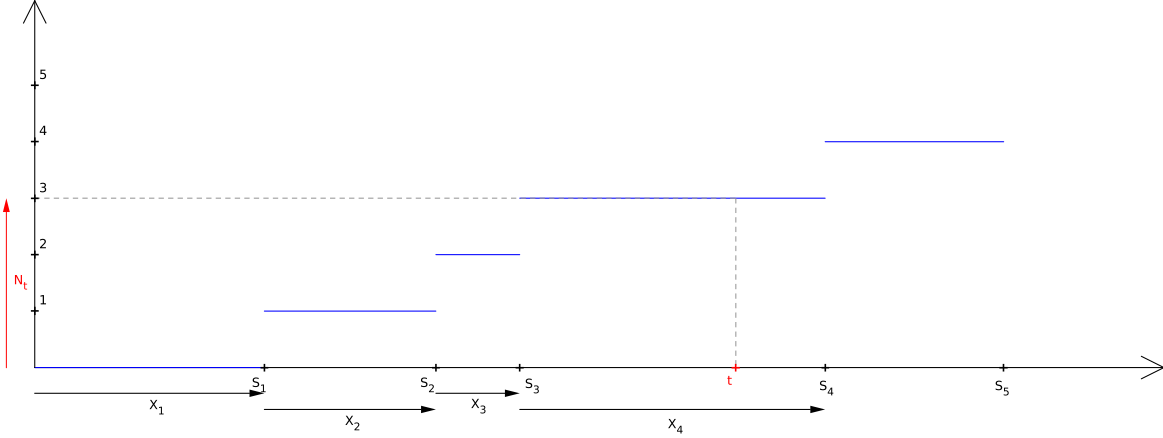


FIGURE 6 – Relations entre  $X_k$ ,  $S_n$  et  $N_t$

Pour étudier les temps d'attente, nous aurons besoin de la propriété suivante :

**THÉORÈME 2.** Soit  $(X_n)_{n \geq 1}$  une suite de variables indépendantes de même loi exponentielle de paramètre  $\lambda > 0$ . Soit  $p \in \mathbb{N}$ . On pose, pour tout  $n \in \mathbb{N}$ ,  $S_n = \sum_{k=1}^n X_k$ .

Alors  $(S_1, \dots, S_p)$  admet pour densité la fonction définie par :

$$\forall (s_1, \dots, s_p) \in \mathbb{R}^p, f_{(S_1, \dots, S_p)}(s_1, \dots, s_p) = \lambda^p e^{-\lambda s_p} \mathbf{1}_{s_1 < \dots < s_p}.$$

Inversement, si un vecteur aléatoire  $(S_1, \dots, S_p)$  admet pour densité cette fonction, alors les  $S_k$  s'écrivent comme les sommes cumulées d'une suite de variables indépendantes de même loi exponentielle.

**Remarque 6.** Si  $s_1 < \dots < s_p$ ,  $f_{(S_1, \dots, S_p)}(s_1, \dots, s_p)$  ne dépend que de  $s_p$  !

*Démonstration.* Soit  $g \in \mathcal{C}^\infty(\mathbb{R}^p)$  à support compact.

On va établir le résultat en calculant l'espérance de  $g((S_1, \dots, S_p))$ .

$$\begin{aligned} \mathbb{E}(g(S_1, \dots, S_p)) &= \mathbb{E}(g(X_1, \dots, \sum_{k=1}^p X_k)) \\ &= \mathbb{E}(g \circ \varphi(X_1, \dots, X_p)) \end{aligned}$$

où  $\varphi : (x_1, \dots, x_p) \in \mathbb{R}^p \mapsto (x_1, x_1 + x_2, \dots, x_1 + \dots + x_p) \in \mathbb{R}^p$  est de classe  $\mathcal{C}^\infty$

Les variables  $X_i$  sont indépendantes de lois à densité, donc  $(X_1, \dots, X_p)$  admet pour densité le produit des densités des  $X_i$ . D'après le théorème de transfert :

$$\mathbb{E}(g(S_1, \dots, S_p)) = \int_{\mathbb{R}^p} g \circ \varphi(x_1, \dots, x_p) \lambda e^{-\lambda x_1} \mathbf{1}_{0 < x_1} \dots \lambda e^{-\lambda x_p} \mathbf{1}_{0 < x_p} dx_1 \dots dx_p$$

$\varphi$  est une bijection de classe  $\mathcal{C}^1$ , on peut donc poser le changement de variable en dimension  $p$ ,  $(u_1, \dots, u_p) = \varphi(x_1, \dots, x_p)$ .

Et  $\varphi^{-1} : (u_1, \dots, u_p) \in \mathbb{R}^p \mapsto (u_1, u_2 - u_1, \dots, u_p - u_{p-1}) \in \mathbb{R}^p$ , donc  $|\text{Jac}(\varphi^{-1})| = 1$ .

Finalement :

$$\mathbb{E}(g(S_1, \dots, S_p)) = \int_{\mathbb{R}^p} g(u_1, \dots, u_p) \lambda^p e^{-\lambda u_p} \mathbf{1}_{0 < u_1 < \dots < u_p} du_1 \dots du_p$$

Donc  $(S_1, \dots, S_p)$  admet pour densité la fonction  $(s_1, \dots, s_p) \in \mathbb{R}^p \mapsto \lambda^p e^{-\lambda s_p} \mathbb{1}_{s_1 < \dots < s_p}$ , ce qui montre l'implication directe.

La démonstration de l'implication réciproque se fait de la même façon, en calculant l'espérance de  $(S_1, S_2 - S_1, \dots, S_p - S_{p-1})$ , puis en faisant le changement de variable inverse. □

Après le théorème précédent, il n'y a plus beaucoup de suspens...

**THÉORÈME 3.** Soit  $(N_t)_{t \in \mathbb{R}_+}$  un processus de Poisson d'intensité  $\lambda > 0$ .

Il existe une suite de variables aléatoires indépendantes  $(X_n)_{n \geq 1}$  suivant la loi exponentielle de paramètre  $\lambda$  telles que si on pose

$$S_0 = 0 \quad \text{et} \quad \forall n \geq 1, \quad S_n = \sum_{k=1}^n X_k$$

alors le processus  $(N_t)_{t \in \mathbb{R}_+}$  vérifie :  $\forall n \in \mathbb{N}, \forall t \in \mathbb{R}_+, \quad N_t = n \Leftrightarrow S_n \leq t < S_{n+1}$ .

*Démonstration.* On définit le processus  $(S_n)_{n \in \mathbb{N}}$  par :

$$S_0 = 0 \quad \text{et} \quad \forall n \in \mathbb{N}, \forall t \in \mathbb{R}_+^*, \quad S_n = t \Leftrightarrow [N_t = n \text{ et } (\forall s < t, N_s < n)]$$

et la suite de variables aléatoires  $(X_n)_{n \geq 1}$  par :

$$\forall k \geq 1, \quad X_k = S_k - S_{k-1}.$$

Il s'agit de montrer que les variables  $X_k$  ainsi définies sont indépendantes et suivent la même loi exponentielle de paramètre  $\lambda$ .

Soit  $m \in \mathbb{N}$ . Soient  $(s_1, \dots, s_m) \in (\mathbb{R}_+)^m$  et  $(h_1, \dots, h_m) \in \mathbb{R}^m$ .

Calculons  $\mathbb{P}((S_1, \dots, S_m) \in ]s_1, s_1 + h_1] \times \dots \times ]s_m, s_m + h_m])$ .

On a :

$$\begin{aligned} & \{(S_1, \dots, S_m) \in ]s_1, s_1 + h_1] \times \dots \times ]s_m, s_m + h_m]\} \\ &= \{N_{s_1} = 0, N_{s_1+h_1} = 1, N_{s_2} = 1, N_{s_2+h_2} = 2, \dots, N_{s_m} = m-1, N_{s_m+h_m} \geq m\} \\ &= \{N_{s_1} = 0, (N_{s_1+h_1} - N_{s_1}) = 1, (N_{s_2} - N_{s_1+h_1}) = 0, (N_{s_2+h_2} - N_{s_2}) = 1, \dots, (N_{s_m+h_m} - N_{s_m}) \geq 1\} \end{aligned}$$

donc :

$$\begin{aligned} & \mathbb{P}((S_1, \dots, S_m) \in ]s_1, s_1 + h_1] \times \dots \times ]s_m, s_m + h_m]) \\ &= \mathbb{P}(N_{s_1} = 0, (N_{s_1+h_1} - N_{s_1}) = 1, (N_{s_2} - N_{s_1+h_1}) = 0, (N_{s_2+h_2} - N_{s_2}) = 1, \dots, (N_{s_m+h_m} - N_{s_m}) \geq 1) \\ &= \mathbb{P}(N_{s_1} = 0) \mathbb{P}(N_{s_1+h_1} - N_{s_1} = 1) \mathbb{P}(N_{s_2} - N_{s_1+h_1} = 0) \mathbb{P}(N_{s_2+h_2} - N_{s_2} = 1) \dots \mathbb{P}(N_{s_m+h_m} - N_{s_m} \geq 1) \\ &= e^{-\lambda s_1} (\lambda h_1 e^{-\lambda h_1}) e^{-\lambda(s_2 - s_1 - h_1)} (\lambda h_2 e^{-\lambda h_2}) \dots (1 - e^{-\lambda h_m}) \quad \text{si tous les } h_i \text{ sont positifs} \\ &= e^{-\lambda s_m} \lambda^{m-1} h_1 h_2 \dots h_{m-1} (1 - e^{-\lambda h_m}) \\ &= \lambda^{m-1} h_1 h_2 \dots h_{m-1} \lambda \int_{s_m}^{s_m+h_m} e^{-\lambda u} du \\ &= \int_{s_1}^{s_1+h_1} \int_{s_2}^{s_2+h_2} \dots \int_{s_{m-1}}^{s_{m-1}+h_{m-1}} \int_{s_m}^{s_m+h_m} \lambda^m e^{-\lambda u} du du_{m-1} \dots du_2 du_1 \end{aligned}$$

et si l'un des  $h_i$  ou  $t_i$  est négatif ou nul :

$$\mathbb{P}((S_1, \dots, S_m) \in ]s_1, s_1 + h_1] \times \dots \times ]s_m, s_m + h_m]) = 0.$$

On peut donc écrire :

$$\begin{aligned} \mathbb{P}((S_1, \dots, S_m) \in ]s_1, s_1 + h_1] \times \dots \times ]s_m, s_m + h_m]) \\ = \int_{s_1}^{s_1+h_1} \int_{s_2}^{s_2+h_2} \dots \int_{s_{m-1}}^{s_{m-1}+h_{m-1}} \int_{s_m}^{s_m+h_m} \lambda^m e^{-\lambda u_m} \mathbb{1}_{u_1 < \dots < u_m} du_m du_{m-1} \dots du_2 du_1. \end{aligned}$$

On en déduit que  $(S_1, \dots, S_m)$  admet pour densité  $f_{(S_1, \dots, S_m)}$  définie par :

$$\forall (u_1, \dots, u_m) \in \mathbb{R}^m, f_{(S_1, \dots, S_m)}(u_1, \dots, u_m) = \lambda^m e^{-\lambda s_m} \mathbb{1}_{u_1 < \dots < u_m}$$

On reconnaît la loi de densité d'une somme de variables aléatoires indépendantes de même loi exponentielle de paramètre  $\lambda$  (cf. THÉORÈME 2).

Ce qui montre que les  $X_i$  sont indépendantes et suivent la même loi  $\mathcal{E}(\lambda)$ . □

**Remarque 7.** On aurait aussi pu définir un processus de Poisson par ses temps d'attente, et démontrer ensuite que ce processus respecte les propriétés énoncées ici comme définition. Par exemple, [5] adopte cette présentation. On étudie cette autre construction dans la partie suivante.

On a vu qu'un processus de Poisson se rapproche d'une idée du hasard : la connaissance des événements passés ne donne aucune information sur les événements futurs. Cela se retrouve ici : la loi exponentielle étant sans mémoire, savoir le temps qui s'est écoulé depuis le dernier spike n'apporte aucune information concernant le temps qu'il reste avant le prochain spike.

### 3.4 Une autre construction

On vient de voir que les temps d'attente associés à un processus de Poisson sont des variables aléatoires indépendantes qui suivent la même loi exponentielle. Inversement, une suite de variables aléatoires indépendantes suivant la loi exponentielle de paramètre  $\lambda$  correspond à des temps d'attente d'un processus de Poisson. Autrement dit, on peut définir un processus de Poisson *via* ses temps d'attente.

Pour démontrer cela, établissons tout d'abord la loi suivie par la somme des temps d'attente, c'est-à-dire la loi suivie par les instants où surviennent les événements.

THÉORÈME 4. Soit  $(X_n)_{n \geq 1}$  une suite de variables indépendantes de même loi exponentielle de paramètre  $\lambda > 0$ . Soit  $n \in \mathbb{N}$ ,  $n \geq 1$ .

La variable aléatoire  $S_n = \sum_{k=1}^n X_k$  admet pour densité la fonction définie par :

$$\forall x \in \mathbb{R}, f_n(x) = \frac{(\lambda x)^{n-1}}{(n-1)!} \lambda e^{-\lambda x} \mathbb{1}_{\mathbb{R}_+^*}(x)$$

**Remarque 8.** On note  $\Gamma(n, \lambda)$  la loi de  $S_n$ .

*Démonstration.* On démontre le résultat par récurrence, en utilisant la propriété suivante : la somme de deux variables aléatoires indépendantes  $X$  et  $Y$  à densité  $f_X$  et  $f_Y$  suit une loi à densité, de densité  $f_X * f_Y$ , convolution de  $f_X$  et  $f_Y$ .

Pour  $p = 1$ ,  $S_p = X_1$  suit une loi exponentielle de paramètre  $\lambda$ , de densité  $f_1$ .

Soit  $p \in \mathbb{N}$ ,  $p \geq 1$ . On suppose que pour tout  $1 \leq n \leq p$ ,  $S_n = \sum_{k=1}^n X_k$  admet pour densité  $f_n$ . Alors  $S_{p+1} = S_p + X_{p+1}$  admet  $f_p * f_1$  pour densité, car  $S_p$  et  $X_1$  sont indépendantes et de densité  $f_p$  et  $f_1$  respectivement.

Soit  $t \in \mathbb{R}_+$ .

$$\begin{aligned}
f_p * f_1(t) &= \int_{-\infty}^{+\infty} f_p(t-u)f_1(u)du \\
&= \int_{-\infty}^{+\infty} \frac{(\lambda(t-u))^{p-1}}{(p-1)!} \lambda e^{-\lambda(t-u)} \mathbb{1}_{\mathbb{R}_+^*}(t-u) \lambda e^{-\lambda u} \mathbb{1}_{\mathbb{R}_+^*}(u) du \\
&= \int_0^t \frac{\lambda^{p+1} (t-u)^{p-1}}{(p-1)!} e^{-\lambda t} du \quad (t \geq 0) \\
&= \frac{\lambda^{p+1}}{(p-1)!} e^{-\lambda t} \left[ -\frac{(t-u)^p}{p} \right]_0^t \\
&= \frac{\lambda^{p+1}}{p!} e^{-\lambda t} t^p \\
&= f_{p+1}(t)
\end{aligned}$$

Pour  $t < 0$ , on a :  $\forall u \in \mathbb{R}, \mathbb{1}_{\mathbb{R}_+^*}(t-u)\mathbb{1}_{\mathbb{R}_+^*}(u) = 0$  et alors  $f_p * f_1(t) = 0 = f_{p+1}(t)$ .

Ceci achève la récurrence. □

**THÉORÈME 5.** Soit  $(X_n)_{n \geq 1}$  une suite de variables indépendantes de même loi exponentielle de paramètre  $\lambda > 0$ . On pose

$$S_0 = 0 \quad \text{et} \quad \forall n \geq 1, S_n = \sum_{k=1}^n X_k.$$

Alors le processus de comptage  $(N_t)_{t \in \mathbb{R}_+}$  défini par

$$\forall n \in \mathbb{N}, \forall t \in \mathbb{R}_+, N_t = n \Leftrightarrow S_n \leq t < S_{n+1}$$

est un processus de Poisson.

**Remarque 9.** La définition du processus  $(N_t)_{t \in \mathbb{R}_+}$  est équivalente à la définition suivante :

$$\forall t \in \mathbb{R}_+, N_t = \sum_{k=1}^{+\infty} \mathbb{1}_{S_k \leq t}.$$

*Démonstration.*

On définit bien un processus de comptage. Il s'agit de vérifier que c'est un processus de Poisson.

**Etape 1 :**  $N_t \sim \mathcal{P}(\lambda t)$

Soit  $t \in \mathbb{R}_+$ . Soit  $n \in \mathbb{N}$ .

On a :

$$\begin{aligned}
\{N_t = n\} &= \{S_n \leq t\} \cap \{S_{n+1} > t\} \\
&= \{S_n \leq t\} \setminus \{S_{n+1} \leq t\} \quad \text{car} \quad \overline{\{S_{n+1} > t\}} = \{S_{n+1} \leq t\} \subset \{S_n \leq t\}
\end{aligned}$$

donc :

$$\begin{aligned}
\mathbb{P}(N_t = n) &= \mathbb{P}(S_n \leq t) - \mathbb{P}(S_{n+1} \leq t) \\
&= \int_0^t \left( \frac{(\lambda x)^{n-1}}{(n-1)!} - \frac{(\lambda x)^n}{n!} \right) \lambda e^{-\lambda x} dx \\
&= \left[ \frac{(\lambda x)^n}{n!} e^{-\lambda x} \right]_0^t \\
&= \frac{(\lambda t)^n}{n!} e^{-\lambda t}
\end{aligned}$$

c'est-à-dire  $N_t \sim \mathcal{P}(\lambda t)$ .

**Etape 2 :**

Soient  $(t_k)_{k \geq 1}$  une suite croissante de réels positifs et  $(n_k)_{k \geq 1}$  une suite d'entiers.  
Soit  $m \geq 2$  un entier.

Montrons que :

$$\begin{aligned} \mathbb{P}(N_{t_1} = n_1 \cap (N_{t_2} - N_{t_1}) = n_2 \cap \dots \cap (N_{t_m} - N_{t_{m-1}}) = n_m) \\ = \frac{e^{-\lambda t_1} (\lambda t_1)^{n_1}}{n_1!} \frac{e^{-\lambda(t_2-t_1)} (\lambda(t_2-t_1))^{n_2}}{n_2!} \dots \frac{e^{-\lambda(t_m-t_{m-1})} (\lambda(t_m-t_{m-1}))^{n_m}}{n_m!}. \end{aligned} \quad (*)$$

On a :

$$\begin{aligned} & \{N_{t_1} = n_1 \cap (N_{t_2} - N_{t_1}) = n_2 \cap \dots \cap (N_{t_m} - N_{t_{m-1}}) = n_m\} \\ & = \{N_{t_1} = n_1 \cap N_{t_2} = n_1 + n_2 \cap \dots \cap N_{t_m} = n_1 + \dots + n_m\} \\ & = \{S_{n_1} \leq t_1 < S_{n_1+1} \cap S_{n_1+n_2} \leq t_2 < S_{n_1+n_2+1} \cap \dots \cap S_{n_1+\dots+n_m} \leq t_m < S_{n_1+\dots+n_m+1}\} \\ & = \{S_1 < \dots < S_{n_1} \leq t_1 < S_{n_1+1} < \dots < S_{n_1+n_2} \leq t_2 < \dots \leq t_m < S_{n_1+\dots+n_m+1}\}. \end{aligned}$$

Donc, d'après la loi de  $(S_1, S_2, \dots, S_p)$  (cf. THÉORÈME 2) avec  $p = n_1 + \dots + n_m + 1$  :

$$\begin{aligned} \mathbb{P}(N_{t_1} = n_1 \cap (N_{t_2} - N_{t_1}) = n_2 \cap \dots \cap (N_{t_m} - N_{t_{m-1}}) = n_m) \\ = \underbrace{\int_0^{t_1} \dots \int_0^{t_1}}_{n_1 \text{ fois}} \underbrace{\int_{t_1}^{t_2} \dots \int_{t_1}^{t_2}}_{n_2 \text{ fois}} \dots \underbrace{\int_{t_{m-1}}^{t_m} \dots \int_{t_{m-1}}^{t_m}}_{n_m \text{ fois}} \int_{t_m}^{+\infty} \lambda^p e^{-\lambda u_p} \mathbb{1}_{u_1 < \dots < u_p} du_p du_{p-1} \dots du_1 \\ = \underbrace{\int_0^{t_1} \dots \int_0^{t_1}}_{n_1 \text{ fois}} \underbrace{\int_{t_1}^{t_2} \dots \int_{t_1}^{t_2}}_{n_2 \text{ fois}} \dots \underbrace{\int_{t_{m-1}}^{t_m} \dots \int_{t_{m-1}}^{t_m}}_{n_m \text{ fois}} \lambda^{p-1} e^{-\lambda t_m} \mathbb{1}_{u_1 < \dots < u_{p-1}} du_{p-1} \dots du_1. \end{aligned}$$

Il s'agit de calculer la valeur de l'intégrale  $I_k^{(a,b)} = \int_a^b \dots \int_a^b \mathbb{1}_{u_1 < \dots < u_k} du_k \dots du_1$ .

On sait d'une part que

$$\sum_{\sigma \in \mathfrak{S}_k} \mathbb{1}_{u_{\sigma(1)} < \dots < u_{\sigma(k)}} = \mathbb{1}_{[a,b]^k} \text{ presque partout} \quad (\#)$$

car l'égalité est fautive si et seulement si l'un des  $u_i$  est égal à un  $u_j$  pour  $i \neq j$ , ce qui représente un espace de dimension au plus  $k-1$ , donc de volume nul pour la mesure de Lebesgue sur  $\mathbb{R}^k$ .

Et d'autre part

$$\forall \sigma \in \mathfrak{S}_k, \int_a^b \dots \int_a^b \mathbb{1}_{u_{\sigma(1)} < \dots < u_{\sigma(k)}} du_k \dots du_1 = \int_a^b \dots \int_a^b \mathbb{1}_{u_1 < \dots < u_k} du_k \dots du_1 \quad (b)$$

car les points  $(x_1, \dots, x_k)$  de  $[a,b]^k$  tels que  $x_{\sigma(1)} < \dots < x_{\sigma(k)}$  n'occupent ni plus ni moins de volume que ceux vérifiant  $x_1 < \dots < x_k$ . On peut comprendre cela intuitivement comme « il n'y a pas de direction privilégiée ». Plus formellement, il s'agit d'un changement de variable en dimension  $n+1$  :

$$(v_1, \dots, v_k) = \varphi((u_1, \dots, u_k)) = (u_{\sigma(1)}, \dots, u_{\sigma(k)})$$

soit

$$(u_1, \dots, u_k) = \varphi^{-1}((v_1, \dots, v_k)) = (v_{\sigma^{-1}(1)}, \dots, v_{\sigma^{-1}(k)})$$

où l'application  $\varphi$  est bien une bijection de  $[a,b]^k$  dans lui-même de classe  $\mathcal{C}^1$ , et  $|\text{Jac}(\varphi^{-1})| = 1$ .

Comme il y a  $k!$  permutations dans  $\mathfrak{S}_k$ , et que  $\int_a^b \dots \int_a^b \mathbb{1}_{[a,b]^k} du_k \dots du_1 = (b-a)^k$ , on déduit de (#) et (b) que :

$$\boxed{\int_a^b \dots \int_a^b \mathbb{1}_{u_1 < \dots < u_k} du_k \dots du_1 = \frac{(b-a)^k}{k!}}$$

On peut revenir à notre calcul précédent :

$$\begin{aligned} \mathbb{P}(N_{t_1} = n_1 \cap (N_{t_2} - N_{t_1}) = n_2 \cap \dots \cap (N_{t_m} - N_{t_{m-1}}) = n_m) \\ = \underbrace{\int_0^{t_1} \dots \int_0^{t_1}}_{n_1 \text{ fois}} \underbrace{\int_{t_1}^{t_2} \dots \int_{t_1}^{t_2}}_{n_2 \text{ fois}} \dots \underbrace{\int_{t_{m-1}}^{t_m} \dots \int_{t_{m-1}}^{t_m}}_{n_m \text{ fois}} \lambda^{p-1} e^{-\lambda t_m} \mathbb{1}_{u_1 < \dots < u_{p-1}} du_{p-1} \dots du_1 \\ = \lambda^{p-1} e^{-\lambda t_m} \underbrace{\int_0^{t_1} \dots \int_0^{t_1}}_{n_1 \text{ fois}} \mathbb{1}_{u_1 < \dots < u_{n_1}} \underbrace{\int_{t_1}^{t_2} \dots \int_{t_1}^{t_2}}_{n_2 \text{ fois}} \mathbb{1}_{u_{n_1+1} < \dots < u_{n_1+n_2}} \\ \dots \underbrace{\int_{t_{m-1}}^{t_m} \dots \int_{t_{m-1}}^{t_m}}_{n_m \text{ fois}} \mathbb{1}_{u_{n_1+\dots+n_{m-1}+1} < \dots < u_{p-1}} du_{p-1} \dots du_1 \\ = \lambda^{n_1+\dots+n_m} e^{-\lambda t_m} \frac{(t_1)^{n_1}}{n_1!} \frac{(t_2 - t_1)^{n_2}}{n_2!} \dots \frac{(t_m - t_{m-1})^{n_m}}{n_m!} \end{aligned}$$

soit :

$$= \frac{e^{-\lambda t_1} (\lambda t_1)^{n_1}}{n_1!} \frac{e^{-\lambda(t_2-t_1)} (\lambda(t_2-t_1))^{n_2}}{n_2!} \dots \frac{e^{-\lambda(t_m-t_{m-1})} (\lambda(t_m-t_{m-1}))^{n_m}}{n_m!}.$$

Ce qui montre (\*).

**Étape 3** : Les accroissements sont stationnaires et indépendants

Soient  $t_1 < t_2 \in \mathbb{R}_+$ ,  $n \in \mathbb{N}$ .

D'après l'étape 1,  $N_{t_1}$  est presque sûrement finie (car  $N_{t_1} \sim \mathcal{P}(\lambda t_1)$ ), donc  $(N_{t_1} = k)_{k \in \mathbb{N}}$  forme un système complet d'événements. En utilisant le résultat de l'étape 2 avec  $m = 2$ , on obtient :

$$\begin{aligned} \mathbb{P}(N_{t_2} - N_{t_1} = n) &= \sum_{k=1}^{+\infty} \mathbb{P}(N_{t_1} = k \cap (N_{t_2} - N_{t_1}) = n) \\ &= \sum_{k=1}^{+\infty} \frac{e^{-\lambda t_1} (\lambda t_1)^k}{k!} \frac{e^{-\lambda(t_2-t_1)} (\lambda(t_2-t_1))^n}{n!} \\ &= \frac{e^{-\lambda(t_2-t_1)} (\lambda(t_2-t_1))^n}{n!}. \end{aligned}$$

Donc  $(N_{t_2} - N_{t_1}) \sim \mathcal{P}(\lambda(t_2 - t_1))$ . D'après l'étape 1, c'est aussi la loi de  $N_{t_2-t_1}$  : les accroissements sont stationnaires.

En revenant à (\*), on en déduit que les accroissements sont indépendants (car la loi conjointe des accroissements est égale au produit des lois marginales).

**Conclusion** :

On a montré les propriétés (1), (2), et (3) de la définition 7 d'un processus de Poisson.  $(N_t)_{t \in \mathbb{R}_+}$  est bien un processus de Poisson. □

### 3.5 Généralisation : processus non stationnaire

En pratique, la fréquence d'émission de potentiels d'action d'un neurone varie au cours du temps, par exemple selon les sollicitations extérieures ou encore la tâche que le sujet effectue. On donne simplement la définition suivante, qui rend possible une adaptation du processus de Poisson au cas où la fréquence varie au cours du temps.

**Définition 11.** Soit  $\lambda : \mathbb{R}_+ \mapsto \mathbb{R}_+$  une fonction.

On dit que le processus aléatoire  $(N_t)_{t \in \mathbb{R}_+}$  est un processus de Poisson non stationnaire lorsque :

- $\forall s, t \in \mathbb{R}_+^*, \forall r \in [0, t], (N_{t+s} - N_t)$  et  $N_r$  sont indépendants ;
- $\forall t \in \mathbb{R}_+^*, N_t \sim \mathcal{P} \left( \int_0^t \lambda(u) du \right)$ .

**Remarque 10.** Dans le cas où la fonction  $\lambda$  est constante, égale à  $\lambda_0$ ,  $(N_t)_{t \in \mathbb{R}_+}$  est un processus de Poisson stationnaire, d'intensité  $\lambda_0$ .

### 3.6 Modélisation

Pour étudier le fonctionnement des neurones, on peut analyser des données réelles, enregistrées avec des électrodes. On peut aussi simuler des données informatiquement, suivant un modèle choisi. Une partie des chercheur.se.s et étudiant.e.s que j'ai rencontré.e.s pendant mon stage modélisent ainsi des réseaux de neurones, dans lesquels les concentrations ioniques dans chaque neurone vérifient un système d'équations différentielles. L'émission de spike est ensuite déterminée par ces concentrations.

On peut aussi adopter un point de vue purement probabiliste : on modélise les spikes d'un neurone par un processus de Poisson ; c'est ce que j'ai fait pendant mon stage. Il faut donc pouvoir simuler informatiquement un processus de Poisson. Une des méthodes possibles se fonde sur le théorème suivant :

**THÉORÈME 6.** Soient  $t_0 \in \mathbb{R}_+^*, n \in \mathbb{N}, (N_t)_{t \in \mathbb{R}_+}$  un processus de Poisson stationnaire et  $(X_k)_{k \in \mathbb{N}}$  la suite de variables indépendantes de même loi exponentielle associée au processus.

Sous le conditionnement  $N_{t_0} = n$ , les variables aléatoires  $\left( S_k = \sum_{i=1}^n X_i \right)_{1 \leq k \leq n}$  se répartissent dans  $[0, t_0]$  de la même façon que  $n$  variables aléatoires indépendantes suivant la loi uniforme sur  $[0, t_0]$ .

*Démonstration.* Soit  $k \leq n$ . Soit  $0 \leq u \leq t_0$ . On a :

$$\begin{aligned} \mathbb{P}(S_k \leq u < S_{k+1} \text{ et } N_{t_0} = n) &= \mathbb{P}(N_u = k \text{ et } N_{t_0} - N_u = n - k) \\ &= \mathbb{P}(N_u = k) \mathbb{P}(N_{t_0} - N_u = n - k) && \text{(accroissements indépendants)} \\ &= \frac{(\lambda u)^k e^{-\lambda u}}{k!} \frac{(\lambda(t_0 - u))^{n-k} e^{-\lambda(t_0 - u)}}{(n-k)!} && \text{(loi de Poisson)} \\ &= \frac{\lambda^n u^k (t_0 - u)^{n-k} e^{-\lambda t_0}}{k!(n-k)!}. \end{aligned}$$

On en déduit :

$$\begin{aligned} \mathbb{P}(S_k \leq u < S_{k+1} \mid N_{t_0} = n) &= \frac{\mathbb{P}(S_k \leq u \text{ et } N_{t_0} = n)}{\mathbb{P}(N_{t_0} = n)} \\ &= \binom{n}{k} \left( \frac{u}{t_0} \right)^k \left( \frac{t_0 - u}{t_0} \right)^{n-k}. \end{aligned}$$

On reconnaît la loi de répartition de  $n$  variables indépendantes  $(U_i)_{1 \leq i \leq n}$  suivant la loi uniforme sur  $[0, t_0]$ . En effet, pour tout  $1 \leq i \leq n$ ,  $\mathbb{P}(U_i \leq u) = \frac{u}{t_0}$  et il y a  $\binom{n}{k}$  façons de choisir les  $k$  variables qui seront inférieures à  $u$  parmi les  $n$  totales. □

Pour modéliser un processus de Poisson stationnaire, on peut donc, de façon équivalente :

1. tirer  $n$  variables aléatoires indépendantes suivant la même loi exponentielle, et en déduire les valeurs du processus selon la construction du THÉORÈME 5 ;
2. choisir  $n$  selon une loi de Poisson de paramètre  $\lambda t_0$ , tirer  $n$  variables aléatoires indépendantes selon la loi uniforme dans  $[0, t_0]$ , et en déduire les valeurs du processus.

J'ai adopté la deuxième solution pour simuler un processus de Poisson sous Python. La FIGURE 7 ci-dessous présente des processus de Poisson ainsi obtenus.

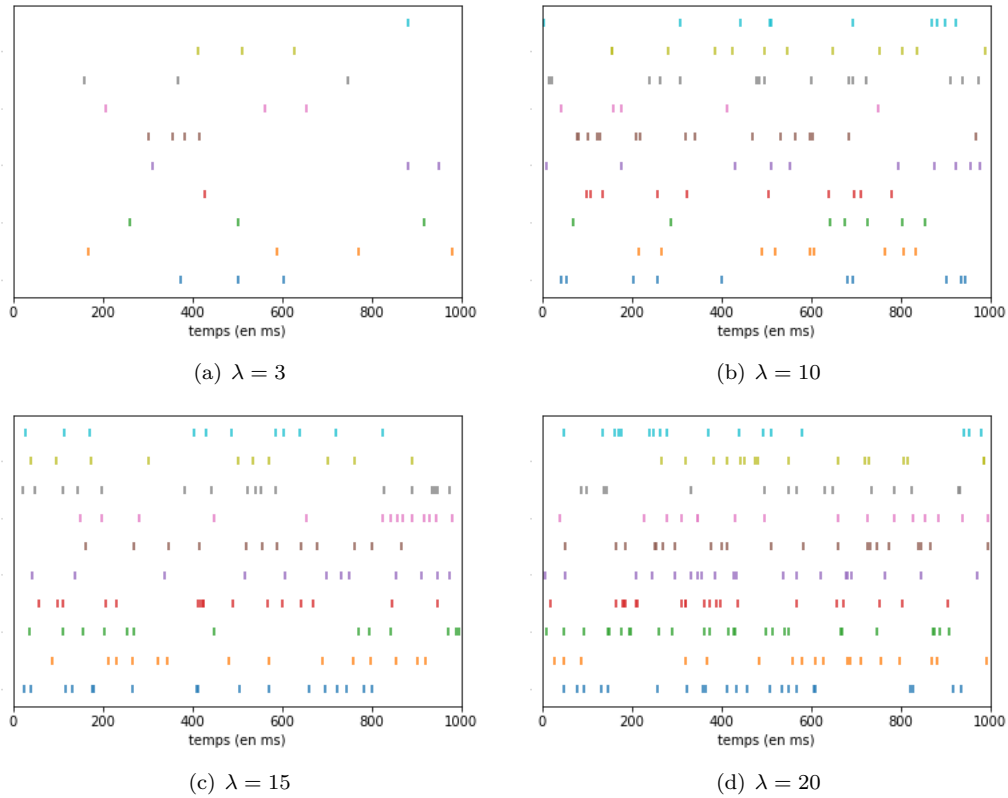


FIGURE 7 – Exemple de simulations de trains de spikes, pour différentes valeurs d'intensité  $\lambda$

Chaque ligne correspond à la simulation d'un processus de Poisson. Les lignes sont indépendantes les unes des autres.

Le théorème précédent se généralise au cas d'un processus de Poisson non stationnaire :

**THÉORÈME 7.** Soient  $t \in \mathbb{R}_+^*$ ,  $n \in \mathbb{N}$ ,  $(N_t)_{t \in \mathbb{R}_+}$  un processus de Poisson non stationnaire d'intensité  $\lambda$  et  $(X_k)_{k \in \mathbb{N}}$  la suite de variables indépendantes de même loi exponentielle associée au processus.

Sous le conditionnement  $N_t = n$ , les variables aléatoires  $\left( S_k = \sum_{i=1}^n X_i \right)_{1 \leq k \leq n}$  se répartissent dans  $[0, t]$  de la même façon que  $n$  variables aléatoires indépendantes suivant la loi de probabilité de fonction de répartition définie par :

$$\forall u \in [0, t], F(u) = \frac{\int_0^u \lambda(y) dy}{\int_0^t \lambda(y) dy}$$

*Démonstration.* La preuve est analogue à la précédente. Soit  $k \leq n$ . Soit  $0 \leq u \leq t$ . On a :

$$\begin{aligned}
\mathbb{P}(S_k \leq u < S_{k+1} \text{ et } N_t = n) &= \mathbb{P}(N_u = k \text{ et } N_t - N_u = n - k) \\
&= \mathbb{P}(N_u = k) \mathbb{P}(N_t - N_u = n - k) && \text{(accroissements indépendants)} \\
&= \frac{\left(\int_0^u \lambda(y) dy\right)^k e^{-\int_0^u \lambda(y) dy}}{k!} \frac{\left(\int_u^t \lambda(y) dy\right)^{n-k} e^{-\int_u^t \lambda(y) dy}}{(n-k)!} && \text{(loi de Poisson)} \\
&= \frac{\left(\int_0^u \lambda(y) dy\right)^k \left(\int_u^t \lambda(y) dy\right)^{n-k} e^{-\int_0^t \lambda(y) dy}}{k!(n-k)!}.
\end{aligned}$$

On en déduit :

$$\begin{aligned}
\mathbb{P}(S_k \leq u < S_{k+1} \mid N_t = n) &= \frac{\mathbb{P}(S_k \leq u \text{ et } N_t = n)}{\mathbb{P}(N_t = n)} \\
&= \binom{n}{k} \frac{\left(\int_0^u \lambda(y) dy\right)^k \left(\int_u^t \lambda(y) dy\right)^{n-k} e^{-\int_0^t \lambda(y) dy}}{\left(\int_0^t \lambda(y) dy\right)^n e^{-\int_0^t \lambda(y) dy}} \\
&= \binom{n}{k} \left(\frac{\int_0^u \lambda(y) dy}{\int_0^t \lambda(y) dy}\right)^k \left(\frac{\int_u^t \lambda(y) dy}{\int_0^t \lambda(y) dy}\right)^{n-k}.
\end{aligned}$$

On reconnaît la loi de répartition de  $n$  variables indépendantes  $(U_i)_{1 \leq i \leq n}$  suivant la loi associée à la fonction de répartition  $F$ . En effet, pour tout  $1 \leq i \leq n$ ,  $\mathbb{P}(U_i \leq u) = \frac{\int_0^u \lambda(y) dy}{\int_0^t \lambda(y) dy}$  et il y a  $\binom{n}{k}$  façons de choisir les  $k$  variables qui seront inférieures à  $u$  parmi les  $n$  totales.

□

Pour modéliser un processus de Poisson non stationnaire, on peut donc tirer le nombre  $n$  de phénomènes aléatoirement selon la loi de Poisson de paramètre  $\int_0^t \lambda(y) dy$ , puis choisir au hasard les instants des événements suivants la fonction de répartition  $F$ .

### Générer un nombre aléatoire suivant une loi de probabilité donnée

On suppose que l'on dispose d'un générateur aléatoire qui fournit  $u \in [0, 1[$  suivant une loi uniforme. À partir de ce générateur, on peut construire un générateur qui suit une autre loi. On présente ici deux techniques générales, on peut imaginer d'autres façons de faire pour certaines lois (par exemple, utiliser le théorème central limite pour générer une loi normale). Cette partie s'inspire de [5], où le lecteur intéressé trouvera plus d'exemples.

#### Méthode de l'inverse

Soit  $\mathcal{L}$  une loi de probabilité, dont on connaît la fonction de répartition  $F$ . On souhaite générer une variable aléatoire  $X$  suivant la loi  $\mathcal{L}$ .

On prend  $u \in [0, 1[$  avec une loi uniforme grâce à notre générateur aléatoire.

On renvoie  $t = \inf_{x \in \mathbb{R}} \{F(x) \geq u\}$ , qui est bien défini car  $F \xrightarrow{+\infty} 1$  donc  $\{F(x) \geq u\}$  est non vide.

Si  $t \leq a$ , alors  $u \leq F(t)$  par définition et  $F(t) \leq F(a)$  par croissance de  $F$ .

Inversement, si  $u \leq F(a)$  alors  $a \in \{F(x) \geq u\}$  donc  $t \leq a$ .

D'où :

$$\mathbb{P}(t \leq a) = \mathbb{P}(u \leq F(a)) = F(a) \quad \text{car } u \sim \mathcal{U}([0, 1])$$

Le réel aléatoire  $t$  renvoyé suit donc effectivement la loi  $\mathcal{L}$ .

**Remarque 11.** Si  $F$  est continue,  $t = \min_{x \in \mathbb{R}} \{F(x) = u\}$ . Si  $F$  est continue et strictement croissante,  $F$  réalise une bijection de  $\mathbb{R}$  dans  $]0, 1[$ , elle est donc inversible et  $t = F^{-1}(u)$  (cette expression n'est pas définie pour  $u = 0$  mais  $u \neq 0$   $\mathbb{P}$ -presque sûrement donc il n'y a pas de problème).

Cette méthode nécessite de connaître le pseudo-inverse de  $F$ , ou de pouvoir le calculer *efficacement*. Si ce n'est pas le cas, on peut suivre la méthode de rejet.

#### Méthode de rejet

Soit  $X$  une variable aléatoire admettant une densité  $f$ .

Simuler  $X$  revient à choisir « uniformément » un point dans le domaine du plan délimité par l'axe des abscisses et la courbe de  $f$ , et ne conserver que son abscisse. On peut aussi choisir  $X$  au hasard dans un domaine plus grand (contenant l'aire sous la courbe de  $f$ ) et ne l'accepter que s'il est en fait sous la courbe de  $f$ . L'intérêt est de choisir un domaine plus grand qui corresponde à l'aire sous la courbe de  $cg$  où  $c > 1$  est un réel et  $g$  une fonction de densité que l'on sait simuler informatiquement. Dire que le domaine est plus grand, c'est dire que  $\forall x \in \mathbb{R}, f(x) \leq cg(x)$ .

Pour tirer aléatoirement un point dans le « grand » domaine (l'aire sous la courbe de  $cg$ ), il suffit de choisir aléatoirement son abscisse  $x_0$  suivant la loi de densité  $g$ , puis de choisir son ordonnée  $y_0$  suivant la loi uniforme sur  $[0, cg(x_0)]$ . Si  $y_0 > f(x_0)$ , on rejette ce point, et on retire  $(x_0, y_0)$  dans le « grand domaine » jusqu'à ce que  $y_0 \leq f(x_0)$ . À ce moment-là, on conserve  $x_0$ .  $x_0$  suit alors la loi de densité  $f$ .

La justification de cette méthode est donnée dans [5].

Si  $f$  est bornée et à support borné, disons  $[a, b]$ , alors on peut choisir  $g : x \in \mathbb{R} \mapsto \frac{1}{b-a} \mathbb{1}_{a \leq x \leq b}$  et  $c = \max_{x \in \mathbb{R}} f(x)$ . Bien sûr, si le « grand » domaine est beaucoup plus grand que l'aire sous la courbe de  $f$ , alors on risque de devoir tirer de nombreux points  $(x_0, y_0)$  avant qu'un point soit accepté. Plus  $cg$  est proche de  $f$ , plus la méthode est efficace.

**Remarque 12.** Matlab et Python proposent des générateurs pour différentes lois. Par exemple, sous Matlab, j'ai pu utiliser un générateur de loi de Poisson sans avoir à le programmer avec les méthodes présentées ci-dessus.

#### Petit aparté sur le pseudo-aléatoire

On a supposé qu'on avait un générateur de nombres aléatoires. Pourtant, générer un nombre aléatoire n'est pas une tâche évidente. En effet, les ordinateurs que nous utilisons sont contrôlés par des programmes déterministes, leurs résultats sont donc *par définition* prévisibles. Et on aimerait obtenir un nombre aléatoire, c'est-à-dire imprévisible...

Les détails des programmes générateurs de pseudo-aléatoire sortent du cadre de ce stage, mais voici quelques aspects qui permettent de se rendre compte du problème.

Les générateurs pseudo-aléatoires utilisés reposent souvent sur des suites récurrentes d'ordre 1, réduites modulo  $m$ , où  $m$  est un nombre premier. Le premier terme de la suite est calculé en fonction de l'horloge de l'ordinateur, ce qui favorise le « hasard » puisqu'en pratique, on ne sait pas quand exactement sera exécuté le programme. Pour simuler le hasard de façon satisfaisante, il faut s'assurer que la période de la suite sera grande, quelle que soit la valeur initiale de la suite. En effet, si la suite ne renvoie que quelques valeurs en boucle, la distribution sera violemment non uniforme. Ce n'est qu'un exemple de difficulté parmi d'autres.

Les derniers générateurs proposés, utilisés par Matlab et Python, se basent sur les nombres de Mersenne, et sont très fiables. On peut réaliser plusieurs millions de tirages successifs, qui pourront être considérés comme indépendants et uniformément distribués.

## 4 Analyse statistique

Maintenant que nous avons développé un modèle probabiliste, nous allons pouvoir revenir à nos questions de départ : connaissant deux trains de spikes, peut-on dire qu'ils sont corrélés ou indépendants ? Quelle signification donner à une distance entre deux trains de spikes ?

Pour y répondre, nous avons besoin de quelques notions de statistiques.

## 4.1 Tests statistiques

Un test statistique se place dans le cadre d'une hypothèse à infirmer. On fait une hypothèse, appelée *hypothèse nulle* et notée  $\mathcal{H}_0$ , et on veut savoir si les résultats obtenus sont crédibles ou s'ils incitent à rejeter  $\mathcal{H}_0$ . Tout l'enjeu du test statistique est de déterminer un seuil à partir duquel le résultat de l'expérience peut être jugé anormal. Par exemple, on lance 10 fois une pièce et on se demande si elle est truquée. On ne s'attend pas à obtenir exactement 5 piles et 5 faces, à partir de quel déséquilibre dira-t-on que la pièce est truquée ? Est-ce que 3 faces et 7 piles est anormal ?

On se place sous l'hypothèse  $\mathcal{H}_0 = \text{« la pièce est équilibrée »}$ . On calcule la probabilité d'obtenir au moins 7 fois pile parmi 10 lancers, en supposant  $\mathcal{H}_0$ , et si la probabilité est plus faible qu'un seuil de tolérance fixé  $\alpha$ , par exemple 0.05, on conclut que la pièce est déséquilibrée en faveur de pile, avec un risque de se tromper inférieur à 5%.

Ici,  $\mathbb{P}_{\mathcal{H}_0}(\text{nombre de piles} \geq 7) = \sum_{k=7}^{10} \binom{10}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{10-k} = 0.171875 > 0.05$  donc l'expérience ne permet pas de dire que la pièce est truquée. Comme dans un tribunal où l'accusé est présumé innocent, la pièce est jugée non coupable, faute de preuves accablantes ([6]).

Après cet exemple introductif, définissons plus précisément quelques caractéristiques d'un test statistique.

**Définition 12** (risques). *On appelle risque de première espèce la probabilité de rejeter  $\mathcal{H}_0$  à tort, et risque de seconde espèce la probabilité de ne pas la rejeter, alors qu'elle est fausse.*

		résultat du test	
		$\mathcal{H}_0$ rejetée	$\mathcal{H}_0$ non rejetée
réalité	$\mathcal{H}_0$ vraie	risque $\alpha$	✓
	$\mathcal{H}_0$ fausse	✓	risque $\beta$

Le risque de première espèce ( $\alpha$ ) correspond au risque de faux positif (on juge la pièce coupable à tort), et le risque de seconde espèce ( $\beta$ ) correspond au faux négatif (on juge la pièce innocente à tort). On maîtrise le risque de première espèce grâce au seuil fixé, mais le risque de seconde espèce est inconnu : on ne connaît souvent même pas la loi de probabilité sous  $\mathcal{H}_1$  ! Par exemple, dans l'exemple de notre pièce,  $\mathcal{H}_1 = \text{« la pièce est déséquilibrée en faveur de pile »}$ , mais on ne sait pas de combien la pièce est déséquilibrée : on ne connaît pas le paramètre de la loi binomiale.

**Définition 13** (puissance d'un test). *On appelle puissance du test la quantité  $1 - \beta$ .*

La puissance du test indique la fiabilité du résultat « on rejette  $\mathcal{H}_0$  » : plus la puissance est élevée, plus la probabilité de rejeter  $\mathcal{H}_0$  à tort est faible.

**Définition 14** (p-valeur). *On appelle p-valeur la probabilité d'obtenir, sous  $\mathcal{H}_0$ , le résultat de l'expérience, ou un résultat plus extrême.*

## 4.2 Tester la synchronisation

Dans le cas de l'étude, on teste  $\mathcal{H}_0 = \text{« les deux processus sont indépendants »}$  contre  $\mathcal{H}_1 = \text{« les deux processus ne sont pas indépendants »}$ . On regarde si le nombre de coïncidences (c'est-à-dire deux spikes qui surviennent dans un même laps de temps) est anormalement élevé, c'est-à-dire avec une probabilité très faible de se produire si  $\mathcal{H}_0$  est vraie. À ce moment-là, on conclut qu'au vu de l'expérience, les deux trains de spikes ne sont pas indépendants. Plus précisément, on note la synchronie que l'on observe entre les deux trains de spikes. Soit  $p$  la probabilité que deux trains de spikes indépendants soient au moins autant synchronisés que notre observation. Si  $p < 0.05$ , on rejette  $\mathcal{H}_0$  et on conclut que les deux trains de spikes sont corrélés.

Cela nécessite donc de se placer dans un cadre bien défini : il faut savoir quelle loi suit le processus pour pouvoir considérer ce qui s'est passé comme étant du « hasard » ou relevant d'une corrélation. Autrement dit, il faut connaître la loi de probabilité sous  $\mathcal{H}_0$ .

On se place dans le cadre présenté dans la partie précédente : on suppose que les trains de spikes sont des processus de Poisson stationnaires, d'intensité  $\lambda$ .

Soient  $X$  et  $Y$  deux trains de spikes, d'une durée de  $T$  millisecondes. On suppose que l'enregistrement est précis à la milliseconde, c'est-à-dire que chaque 0 ou 1 de  $X$  et  $Y$  correspond à une milliseconde.

On calcule la distance entre  $X$  et  $Y$  (cf. partie 2 sur les distances de synchronie), et on veut en déduire si les deux processus sont indépendants. Pour cela, il faut connaître la loi que suit la distance entre deux processus indépendants de Poisson d'intensité  $\lambda$ , sur une durée de  $T$  millisecondes. Dans le cadre de certains modèles, on peut déterminer cette loi analytiquement, ici on l'estimera avec une loi empirique obtenue avec des simulations.

Plusieurs possibilités s'offrent à nous.

On peut simuler un grand nombre de fois deux processus de Poisson de même intensité  $\lambda$  sur une durée de  $T$  millisecondes, calculer la distance pour chacun des couples de trains de spikes, puis regarder la distribution des valeurs de distances obtenues.

On peut aussi adapter la loi empirique aux trains de spikes  $X$  et  $Y$  que l'on teste, c'est-à-dire construire le test statistique à partir de  $X$  et  $Y$ , et non à partir de simulations indépendantes de  $X$  et de  $Y$ . Comme  $X$  et  $Y$  suivent un processus de Poisson – c'est en tout cas notre hypothèse – les temps d'attente entre chaque spike sont des variables aléatoires indépendantes de même loi exponentielle. Un train de spike obtenu en mélangeant les temps d'attente de  $Y$  suit donc la même loi, c'est un processus de Poisson stationnaire d'intensité  $\lambda$  (qui n'est bien sûr pas indépendant de  $Y$ ). Notons  $Z$  un tel train de spike. Si  $X$  et  $Y$  sont indépendants, alors la distance entre  $X$  et  $Y$  ou celle entre  $X$  et  $Z$  suivent la même loi de probabilité. Pour décider si  $X$  et  $Y$  sont indépendants ou non, on peut donc comparer la distance entre  $X$  et  $Y$  à la distribution des distances entre  $X$  et des mélanges de  $Y$ .

Pour faire ces tests, il faut pouvoir simuler des trains de spikes indépendants – il suffit de simuler plusieurs processus de Poisson – et des trains de spikes corrélés. Pour simuler deux trains de spikes non indépendants, j'ai procédé de la manière suivante.

1. Je génère un processus de Poisson (cf. partie **3.6 Modélisation**) d'une durée  $T$ , qui représente le premier train de spikes, notons le  $X$ .
2. Je génère le second train de spikes,  $Y$ , à partir de  $X$  :
  - chaque spike est décalé selon une loi normale centrée de variance  $\sigma^2$  ;
  - chaque spike a une probabilité  $p_d$  d'être supprimé ;
  - on ajoute un certain nombre aléatoire  $m \sim \mathcal{B}(T, \frac{p_a}{T})$  de spikes, placés aléatoirement selon la loi uniforme sur  $[0, T]$ .

La FIGURE 8 ci-dessous présente quelques simulations, pour  $T = 1000$  ms,  $\lambda = 10$  Hz,  $p_d = 0.09$ ,  $p_a = 0.9$  et différentes valeurs de  $\sigma^2$ .

Dans toute la suite, on fixe les paramètres suivants :

- durée du train de spikes :  $T = 1000$  ms
- intensité du processus de Poisson :  $\lambda = 10$  Hz
- variance de la loi normale pour décaler chaque spike :  $v = 10$  ms<sup>2</sup>
- probabilité de disparition pour chaque spike :  $p_d = 0.09$
- probabilité d'apparition d'un spike :  $p_a = 0.9$

Pour le test par comparaison avec la distribution empirique des distances, j'ai généré  $n$  couples de trains de spikes indépendants, j'ai calculé la distance pour chaque couple. J'obtiens ainsi une répartition empirique, qui sera la référence pour le test par comparaison. D'autre part je calcule la distance entre  $X$  et  $Y$ . Le test rejette  $\mathcal{H}_0$  s'il y a moins de 5% des distances qui sont inférieures à celle entre  $X$  et  $Y$ . J'ai pris  $n = 10000$  pour la distance de van Rossum et  $n = 100000$  pour la distance de Victor et Purpura.

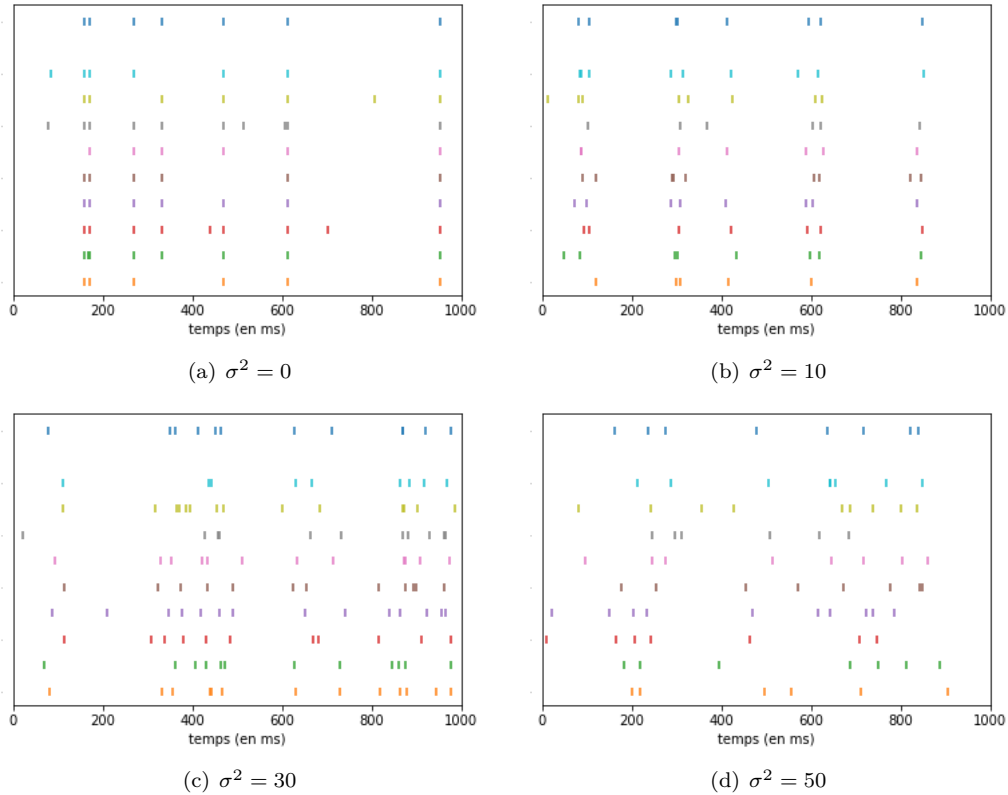


FIGURE 8 – Exemple de simulations de trains de spikes corrélés, pour différentes valeurs de  $\sigma^2$   
 La première ligne correspond à  $X$ , les lignes suivantes correspondent à plusieurs simulations de  $Y$ .

Pour le test par permutations, j'ai généré  $X$  et  $Y$ , et  $n = 10000$  trains de spikes obtenus par mélange des temps d'attente de  $Y$ . Le test rejette  $\mathcal{H}_0$  s'il y a moins de 5% des distances entre  $X$  et chacun des  $n$  trains de spikes qui sont inférieures à celle entre  $X$  et  $Y$ .

J'ai généré des trains des spikes  $X$  et  $Y$  différents pour chaque test.

Avec les paramètres choisis, l'exécution d'un test prend quelques secondes.

Le test par permutation demande de générer  $n$  mélanges de  $Y$  puis de calculer les  $n$  distances à  $X$  à chaque fois qu'on l'utilise. Le test par comparaison demande de générer  $n$  couples de processus de Poisson indépendants et calculer les  $n$  distances qu'une seule fois : quand la distribution empirique est établie, on peut utiliser le test autant de fois que l'on veut. Il faut simplement calculer la distance de  $X$  à  $Y$ , puis la p-valeur.

Cependant, pour utiliser le test par comparaison, il faut connaître la valeur de  $\lambda$ . En pratique, on doit donc d'abord estimer ce paramètre, par exemple en mesurant la fréquence moyenne d'émission de spikes des neurones qu'on étudie.

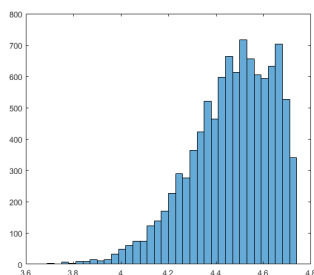
Enfin, le test par permutation est plus puissant (il se "trompe" moins souvent) que le test par comparaison. Rappelons les deux risques de se tromper dans ces tests :

1. il se pourrait que les processus soient indépendants, mais que par hasard ils aient été plutôt synchrones (*risque de première espèce  $\alpha$* );
2. il se pourrait que les processus ne soient pas indépendants, mais que par hasard ils n'aient pas été très synchrones (*risque de seconde espèce  $\beta$* ).

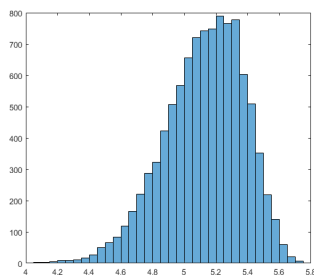
Pour comparer l'efficacité des différents tests, j'ai appliqué plusieurs fois chaque test, avec des trains indépendants et des trains corrélés. J'ai relevé la p-valeur obtenue à chaque fois.

Je présente mes résultats dans les pages suivantes.

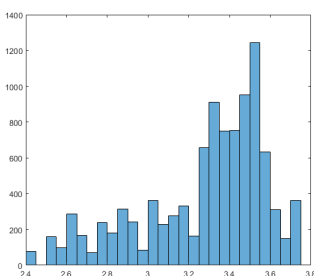
## Tests sous $\mathcal{H}_0$



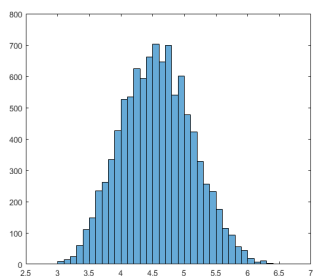
$\tau_R = 5$  ms  
 $D_R(X, Y) = 4.43$   
 $p = 0.38 > 0.05$   
 $\rightsquigarrow$  on accepte  $\mathcal{H}_0$ .



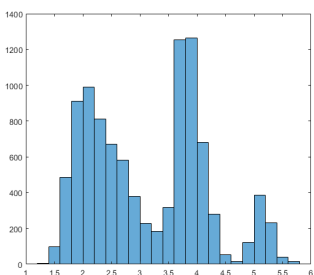
$\tau_R = 10$  ms  
 $D_R(X, Y) = 5.08$   
 $p = 0.43 > 0.05$   
 $\rightsquigarrow$  on accepte  $\mathcal{H}_0$ .



$\tau_R = 50$  ms  
 $D_R(X, Y) = 3.58$   
 $p = 0.89 > 0.05$   
 $\rightsquigarrow$  on accepte  $\mathcal{H}_0$ .

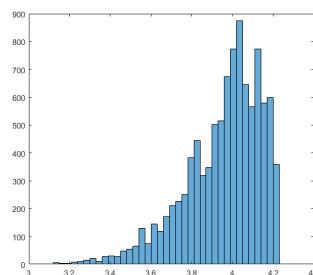


$\tau_R = 100$  ms  
 $D_R(X, Y) = 4.09$   
 $p = 0.2 > 0.05$   
 $\rightsquigarrow$  on accepte  $\mathcal{H}_0$ .

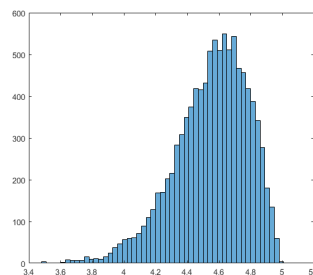


$\tau_R = 500$  ms  
 $D_R(X, Y) = 1.94$   
 $p = 0.1192 > 0.05$   
 $\rightsquigarrow$  on accepte  $\mathcal{H}_0$ .

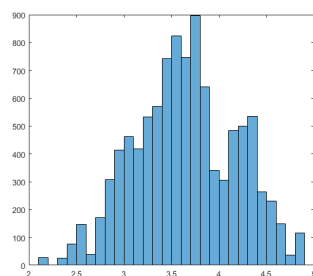
## Tests sous $\mathcal{H}_1$



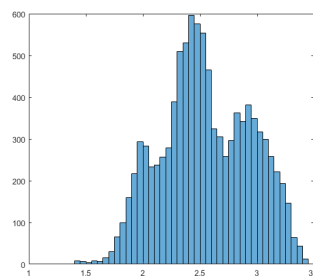
$\tau_R = 5$  ms  
 $D_R(X, Y) = 3.06$   
 $p = 0.0002 < 0.05$   
 $\rightsquigarrow$  on rejette  $\mathcal{H}_0$ .



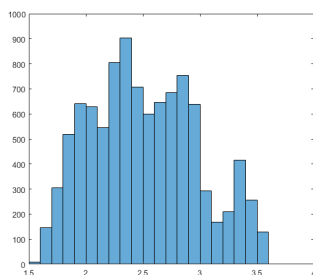
$\tau_R = 10$  ms  
 $D_R(X, Y) = 3.74$   
 $p = 0.0037 < 0.05$   
 $\rightsquigarrow$  on rejette  $\mathcal{H}_0$ .



$\tau_R = 50$  ms  
 $D_R(X, Y) = 2.18$   
 $p = 0.0014 < 0.05$   
 $\rightsquigarrow$  on rejette  $\mathcal{H}_0$ .



$\tau_R = 100$  ms  
 $D_R(X, Y) = 1.72$   
 $p = 0.0064 < 0.05$   
 $\rightsquigarrow$  on rejette  $\mathcal{H}_0$ .

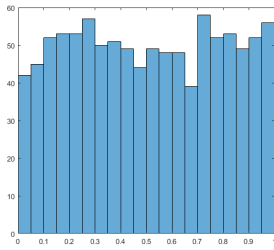


$\tau_R = 500$  ms  
 $D_R(X, Y) = 1.61$   
 $p = 0.003 < 0.05$   
 $\rightsquigarrow$  on rejette  $\mathcal{H}_0$ .

FIGURE 9 – Résultats des tests par permutation, avec la distance de van Rossum pour différentes valeurs de  $\tau_R$

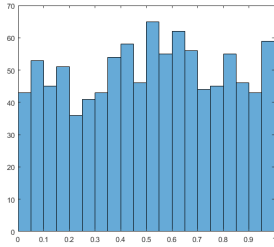
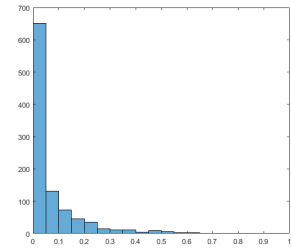
Chaque test a été construit avec 10000 mélanges. Tous les tests présentés ici ont donné une réponse juste. Cela ne montre pas que notre test est puissant! Pour évaluer sa puissance, la figure suivante montre les distributions des p-valeurs dans les différents cas.

Tests sous  $\mathcal{H}_0$

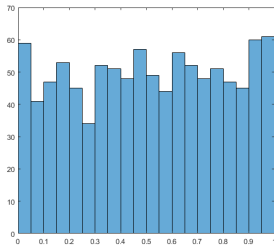
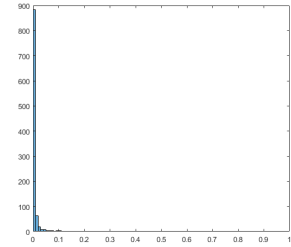


$\tau_R = 5$  ms

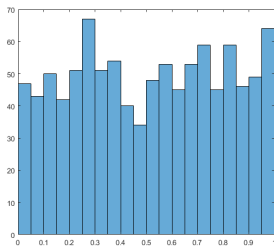
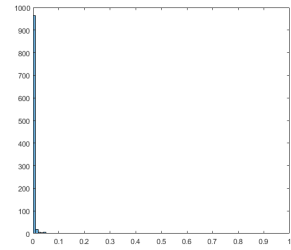
Tests sous  $\mathcal{H}_1$



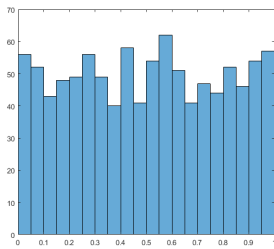
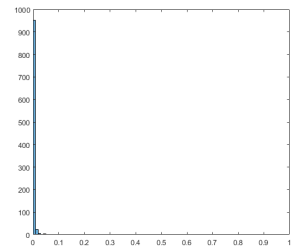
$\tau_R = 10$  ms



$\tau_R = 50$  ms



$\tau_R = 100$  ms



$\tau_R = 500$  ms

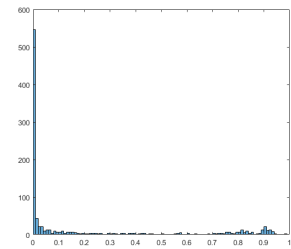


FIGURE 10 – Distributions des p-valeurs lors du test par permutation, avec la distance de van Rossum, pour différentes valeurs de  $\tau_R$

On observe une répartition uniforme sous  $\mathcal{H}_0$ . Le test est puissant pour  $\tau_R = 10, 50, 100$  ms. Pour  $\tau_R = 5, 500$  ms,  $\beta > 0.3$ , on a une grande chance de ne pas détecter une synchronisation ! Dans le premier cas  $\tau_R = 5$  ms, le temps caractéristique est très faible, on est très exigeant sur la synchronisation, la puissance serait sûrement plus élevée avec une corrélation plus forte. Dans le second cas  $\tau_R = 500$  ms, le temps caractéristique est très élevé par rapport à la durée des trains de spike (1000 ms), on assiste donc probablement à des effets de bords.

Le test par permutation est plus puissant avec la distance de Victor et Purpura. En effet, la répartition des p-valeurs est la suivante :

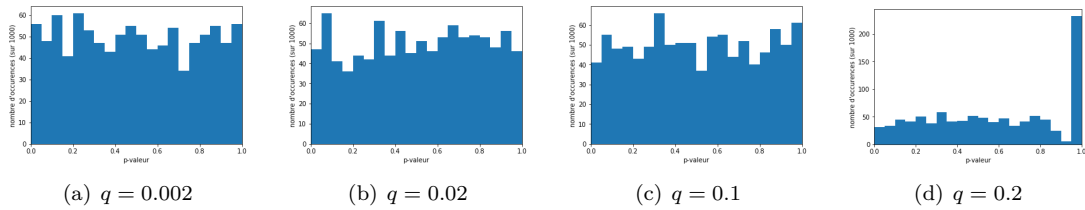


FIGURE 11 – Répartition des p-valeurs sous  $\mathcal{H}_0$  lors du test par permutation, avec la distance de Victor et Purpura

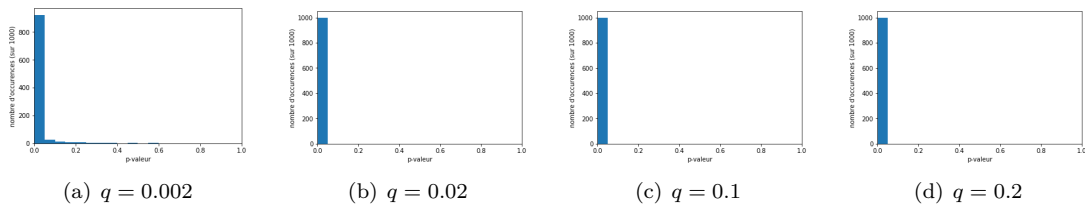


FIGURE 12 – Répartition des p-valeurs sous  $\mathcal{H}_1$  lors du test par permutation, avec la distance de Victor et Purpura

Pour  $q = 0.2$ , la répartition des p-valeurs sous  $\mathcal{H}_0$  n'est pas uniforme, et il y a moins de 5% des essais qui sont jugés (à tort) comme corrélés. Ici, la loi étudiée – la distance de Victor et Purpura entre deux trains de spikes indépendants – est discrète, les p-valeurs ne se répartissent donc pas nécessairement uniformément, contrairement au cas des lois continues. Plus  $q$  est faible, plus les valeurs possibles pour la distance de Victor et Purpura peuvent être proches. On peut alors considérer que la distance est continue, c'est le passage du discret au continu<sup>2</sup>. On visualise effectivement ce passage du discret au continu dans la distribution des distances de Victor et Purpura (cf. FIGURE 13).

2. « phénomène » qui explique par exemple qu'on ne voit pas d'effet quantique à notre échelle

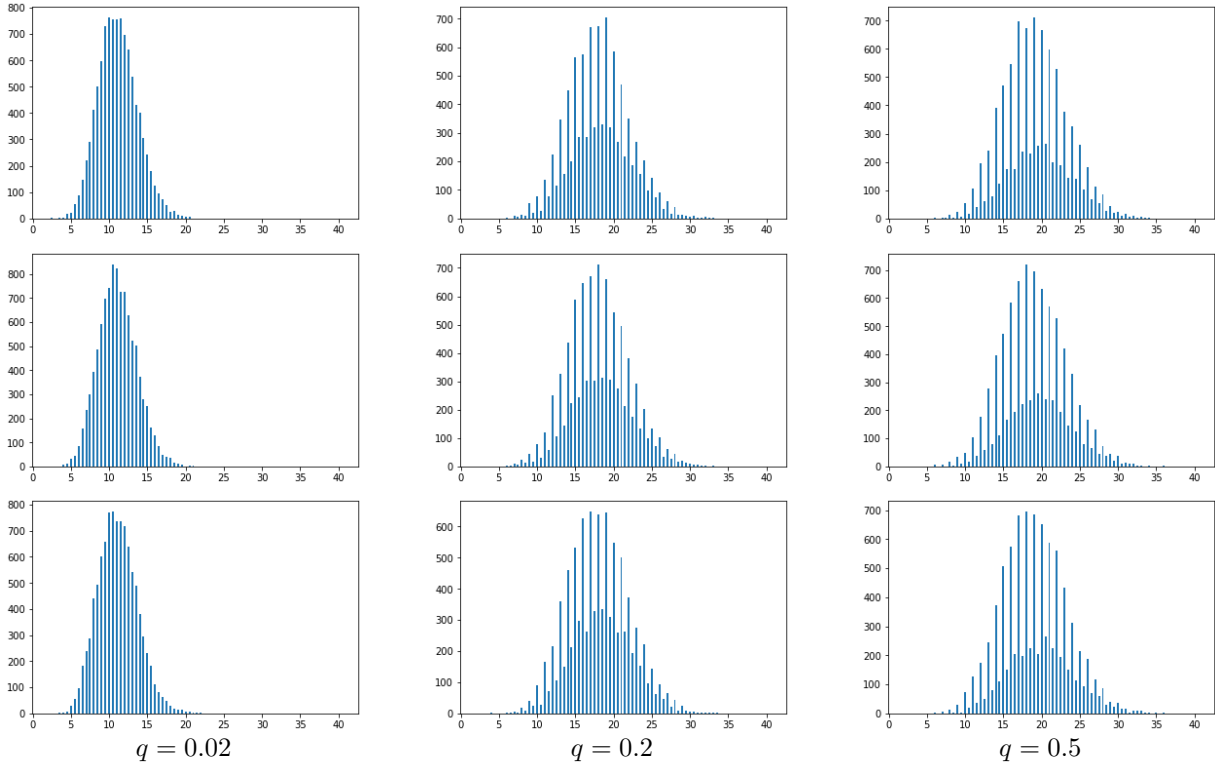


FIGURE 13 – Distribution des distances de Victor et Purpura, sous  $\mathcal{H}_0$ , obtenues par tirages successifs. Chaque colonne correspond à un choix du paramètre  $q$  de la distance de Victor et Purpura. Chaque distribution a été simulée trois fois (une par ligne) pour se rendre compte de la variabilité entre les tirages. Chaque distribution est obtenue avec 10000 couples de trains de spikes. Il faut une à deux secondes pour obtenir chacune des distributions.

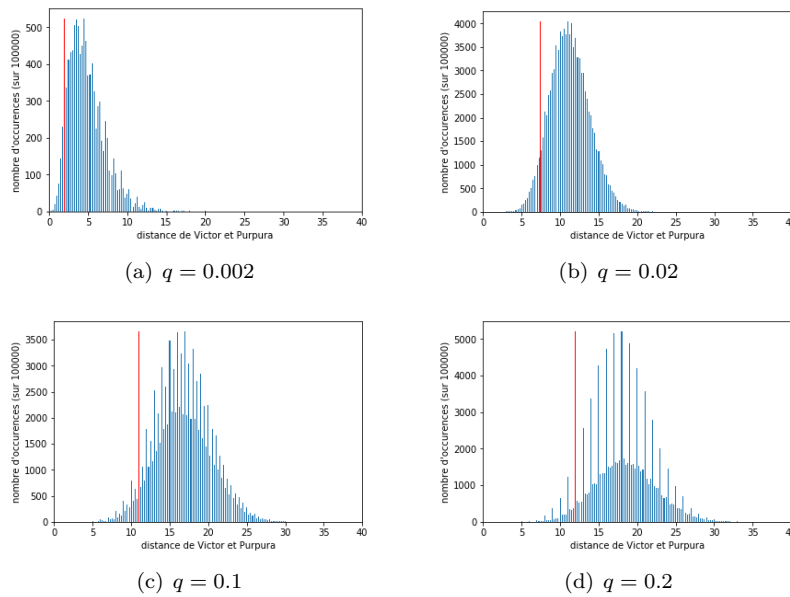


FIGURE 14 – Distribution des distances de Victor et Purpura sous  $\mathcal{H}_0$ , obtenues par tirages successifs. La ligne rouge marque la limite du 5<sup>ième</sup> centile, c'est-à-dire le seuil pour le test. Les p-valeurs obtenues sont présentées figure suivante.

Le test par comparaison est en fait moins efficace que le test par permutation, comme le montrent les histogrammes suivants.

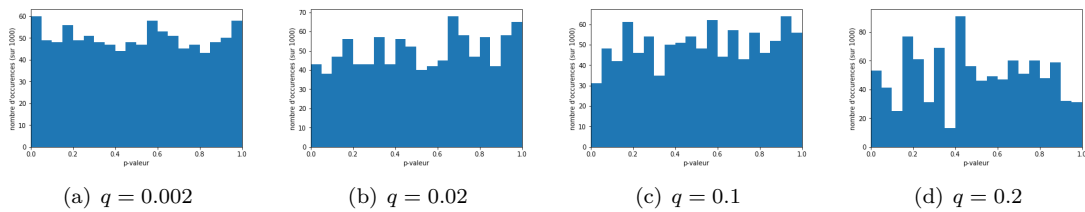


FIGURE 15 – Répartition des p-valeurs sous  $\mathcal{H}_0$  lors du test par comparaison, avec la distance de Victor et Purpura

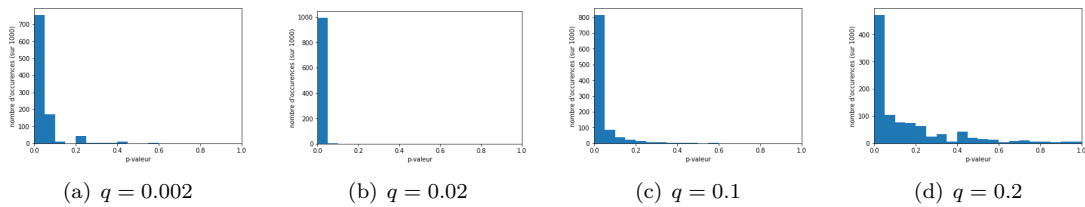


FIGURE 16 – Répartition des p-valeurs sous  $\mathcal{H}_1$  lors du test par comparaison, avec la distance de Victor et Purpura

Ici aussi, les p-valeurs ne sont pas réparties uniformément pour  $q = 0.2$ , cela s'explique de même par le caractère discret de la distribution empirique (contrairement au cas  $q = 0.002$ , où l'on peut considérer que la distribution discrète est en fait continue).

Ces histogrammes nous montrent que le test par comparaison avec la distribution empirique de la distance de Victor et Purpura est très puissant dans le cas  $q = 0.02$ , et beaucoup moins pour les autres valeurs de  $q$ .

Voici les distances obtenues entre un train de spikes  $X$  et plusieurs mélanges du même autre train de spikes  $Y$ . On présente les résultats obtenus pour différents couples  $(X, Y)$ . On remarque que l'allure des courbes dépend surtout du paramètre  $q$  de la distance de Victo et Purpura, et assez peu de  $(X, Y)$ .

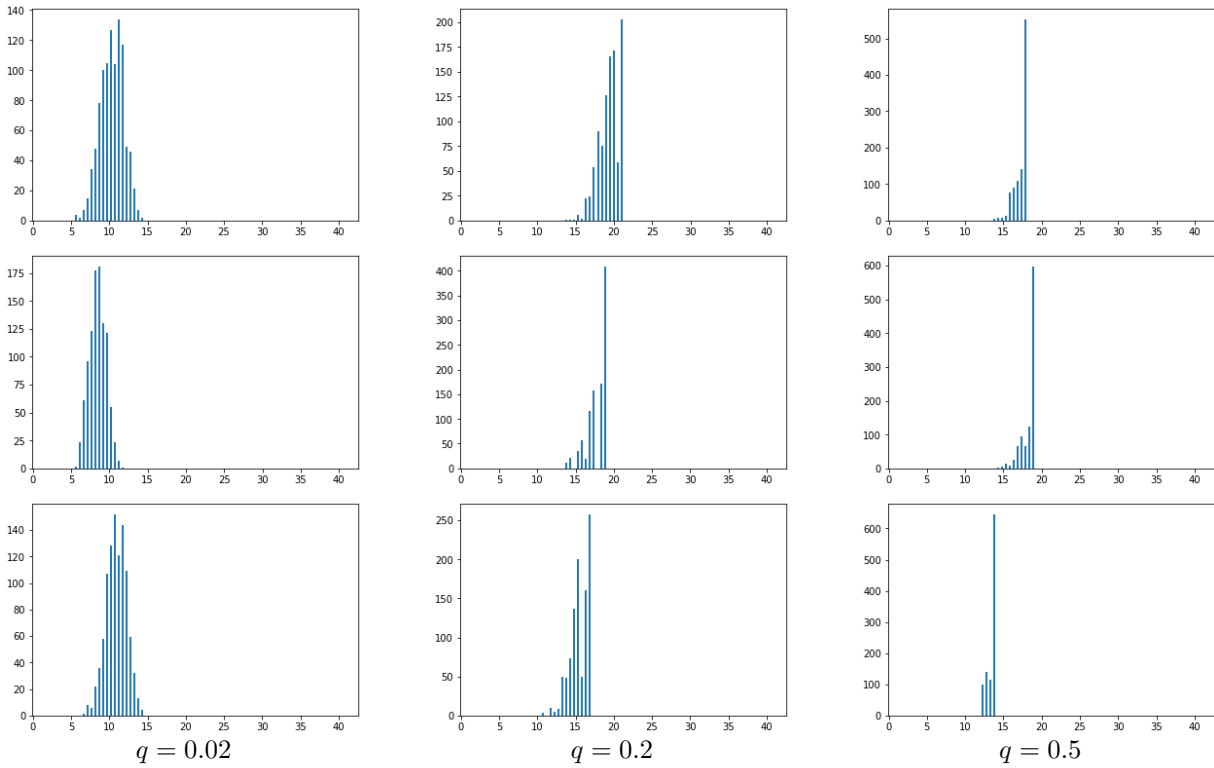


FIGURE 17 – Distribution des distances de Victor et Purpura, sous  $\mathcal{H}_0$ , obtenues par permutations  
 Chaque colonne correspond à un choix du paramètre  $q$  de la distance de Victor et Purpura. Chaque distribution a été simulée trois fois (une par ligne) pour se rendre compte de la variabilité entre les tirages. Chaque distribution est obtenue avec 1000 couples de trains de spikes.

Voici les distributions des distances de van Rossum pour différentes valeurs de  $\tau_R$ . On n'observe pas cette fois-ci le passage du discret au continu comme pour la distance de Victor et Purpura. En effet, la distance de van Rossum peut prendre *a priori* toutes les valeurs réelles, alors que la distance de Victor et Purpura est à valeurs dans l'ensemble discret  $\mathbb{N} + q\mathbb{N}$ .

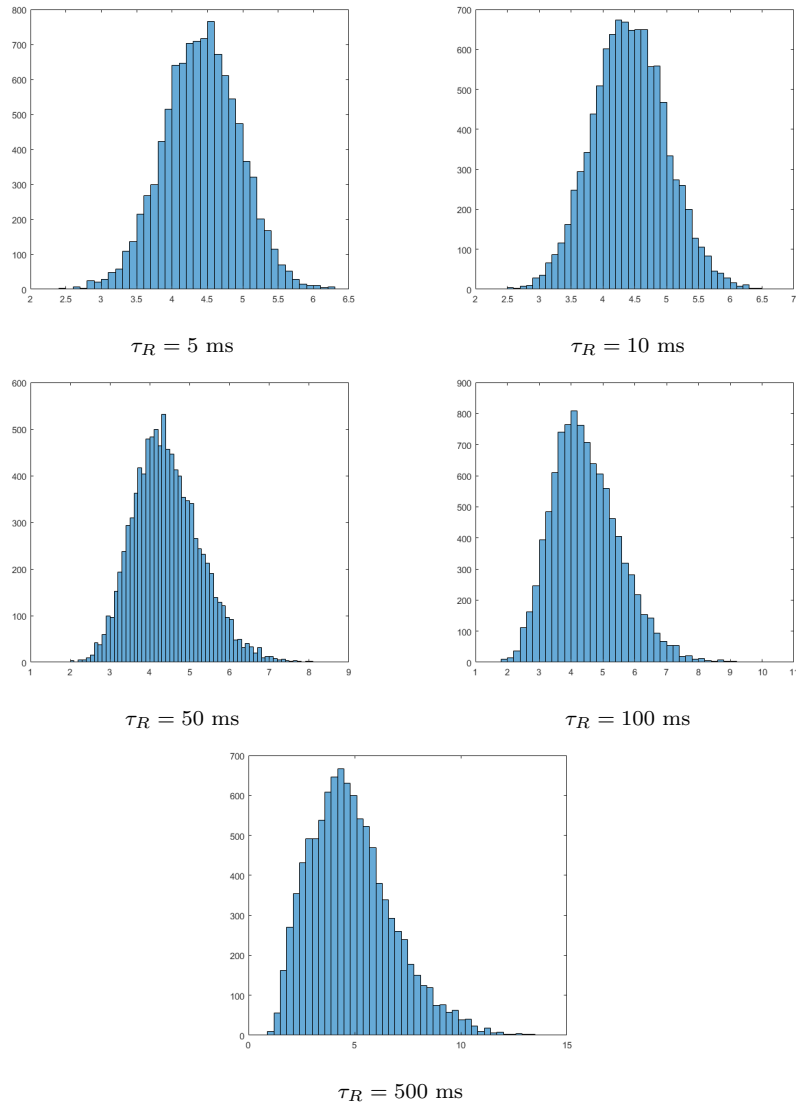


FIGURE 18 – Distribution des distances de van Rossum, sous  $\mathcal{H}_0$ , en fonction du paramètre  $\tau_R$ , obtenues par tirages successifs

En abscisse, la distance de van Rossum ; en ordonnée, le nombre d'occurrences.

Chaque histogramme est obtenu avec 10000 tirages de deux processus de Poisson indépendants, stationnaires d'intensité  $\lambda = 10$  Hz, pendant 1000 ms. Chaque histogramme correspond à un paramètre  $\tau_R$  différent.

On observe que la distance moyenne et la variance augmentent avec  $\tau_R$ .

### 4.3 Conclusion

Pour répondre à la question « deux neurones sont-ils synchrones? », il est essentiel de définir ce que l'on entend par synchronisation. Est-ce que deux processus de Poisson non stationnaires de même intensité  $\lambda(t)$  mais indépendants sont synchrones parce qu'ils sont plus actifs en même temps (les instants de spikes suivent la même répartition) ou bien sont-ils non synchrones car indépendants? Il n'y a pas de réponse *a priori*, cela dépend de l'aspect que l'on cherche à étudier. Seule une discussion entre neurobiologistes et statisticiens peut permettre de choisir la définition la plus pertinente.  $\mathcal{H}_0$  sera choisie en conséquence.

Dans le cadre de ce stage, j'ai étudié le cas du processus de Poisson stationnaire. Ce modèle n'est qu'une approximation grossière de la réalité. En effet, après avoir émis un spike, un neurone est dans une phase de repos pendant quelques millisecondes, c'est la *période réfractaire*. Le modèle du processus de Poisson ne tient pas du tout compte de ce temps de repos. Il est donc essentiel d'étudier d'autres modèles pour plus de réalisme.

D'autres points sont critiquables. Dans son article [10], Sonja Grün pointe deux autres sources d'erreurs. On a supposé que les processus étaient stationnaires, ce qui est rarement le cas : les neurones ont des périodes d'activités plus ou moins intenses. On a également supposé que les résultats des différents essais suivaient le même modèle. Or l'intensité  $\lambda$  peut varier au cours des essais, à cause des conditions expérimentales, dues par exemple à de la fatigue chez le sujet. Ne pas tenir compte de ces différents écarts au modèle du processus de Poisson stationnaire produit des faux positifs ou négatifs.

Dans ce mémoire, j'ai présenté des tests effectués à partir de simulations. Je connaissais donc les paramètres utilisés, ce qui n'est pas le cas en pratique. En particulier, on n'a accès qu'à une valeur empirique du paramètre  $\lambda$ . Il y a donc plusieurs sources d'approximation : le choix du modèle, l'estimation du ou des paramètre(s), l'application du théorème central limite pour le calcul des intervalles de confiance. Tous ces aspects seraient des pistes pour approfondir le sujet exposé ici.

On peut aussi se servir des outils présentés dans ce rapport pour détecter des motifs particuliers qui se répètent lors d'une certaine tâche (par exemple, reconnaître un visage). On peut aussi s'en servir pour déterminer si la réponse d'un neurone est précise rythmiquement : si on répète plusieurs fois la même expérience, est ce que la réponse du neurone (ou de la population de neurones dans son ensemble) est exactement la même? ou bien seule la fréquence est conservée?

Répondre à ces questions permet de mieux comprendre le cerveau, et inspire également la recherche en robotique.

# Bibliographie

- [1] <https://upload.wikimedia.org/wikipedia/commons/e/eb/Neurone.png>.
- [2] [http://passeport.univ-lille1.fr/site/biologie/scbio/Neurone/Neurone\\_web.publi/web/res/03PropriFig24.jpg](http://passeport.univ-lille1.fr/site/biologie/scbio/Neurone/Neurone_web.publi/web/res/03PropriFig24.jpg).
- [3] <http://www-users.med.cornell.edu/~jdvicto/pubalgor.html>.
- [4] <http://wwold.fi.isc.cnr.it/users/thomas.kreuz/Source-Code/VanRossum.html>.
- [5] Walter APPEL. *Probabilités pour les non-probabilistes*. H et K, 2013.
- [6] Benoît CADRE. Le test statistique par l'exemple, complément de cours pour la préparation à l'agrégation externe, 2016-2017.
- [7] Julien CHEVALLIER, Patricia REYNAUD-BOURET, and Marc MONTICELLI. Nos neurones se synchronisent-ils ? Images des Mathématiques, CNRS, 2017.
- [8] Justin DAUWELS, Francois VIALATTE, Theophane WEBER, Toshimitsu MUSHI, and Andrzej CICHOCKI. Quantifying statistical interdependence by message passing on graphs — part I : One-dimensional point processes. 21 :2203–2268, 08 2009.
- [9] Dominique FOATA and Aimé FUCHS. *Processus stochastiques*. Dunod, 2002.
- [10] Sonja GRÜN. Data-driven significance estimation for precise spike correlation. *Journal of Neurophysiology*, 101(3) :1126–1140, 2009. PMID : 19129298.
- [11] Sonja GRÜN, Markus DIEMANN, and Ad AERTSEN. Unitary events in multiple single-neuron spiking activity : I. detection and significance. *Neural computation*, 14 1 :43–80, 2002.
- [12] Matthew T. HARRISON, Asohan AMARASINGHAM, and E. KASS. Statistical identification of synchronous spiking, 2013.
- [13] Conor HOUGHTON and Thomas KREUZ. On the efficient calculation of van rossum distances. *Network : Computation in Neural Systems*, 23(1-2) :48–58, 2012. PMID : 22568695.
- [14] Thomas KREUZ. Measures of spike train synchrony. [http://www.scholarpedia.org/article/Measures\\_of\\_spike\\_train\\_synchrony](http://www.scholarpedia.org/article/Measures_of_spike_train_synchrony).
- [15] Jean-François LE GALL. Intégration, probabilités et processus aléatoires, septembre 2006.