Projet GM2 19-20 orienté Maths/Finance. Simulation du modèle de Lundberg

PROJET : Simulation du modèle de Lundberg

Ruine en Assurance

Groupe 2 – MF01

ZIRIHI YANNICK

KOUROUMA ABOUBACAR

EL KASMI ADNANE

- **Introduction :**

La théorie de la ruine appartient aux sciences de la gestion des risques et aux mathématiques appliquées à l'assurance. Il s'agit de l'étude mathématique de modèles stochastiques et dynamiques adaptés aux réserves financières allouées à un portefeuille de contrats d'assurance de type non-vie d'une compagnie d'assurance. Assurance de type IARD (Incendie, Accidents et Risques Divers).

L'objectif est de définir un cadre permettant la bonne gestion d'un portefeuille de contrat. On imagine une compagnie d'assurance qui se lance sur un nouveau marché, le business line doit être :

- ✓ Viable, la compagnie doit être solvable a tout instant (la réserve ne doit pas tomber en dessous de 0). Elle doit être en mesure de faire face aux engagements qu'elle a pris vis à vis des assurés par voie contractuelle.
- ✓ Rentable, la tarification doit permettre à l'assureur d'engranger des bénéfices pour rémunérer les actionnaires et les employés.

L'étude des processus aléatoires s'insère dans la théorie des probabilités dont elle constitue l'un des objectifs les plus profonds. Elle soulève des problèmes mathématiques intéressants et souvent très difficiles. Ce projet présente quelques aspects des processus aléatoires utiles à l'ingénieur mathématicien du fait de leur fréquence d'occurrence dans les applications : processus de renouvellement, processus de Markov, mouvement brownien etc.

Livrable 1. - Activité 1. Etudes mathématiques.

Cet activité vise à vous familiariser avec des notions mathématiques de l'assurance et avec des bases de la théorie de la ruine : le modèle de Lundberg-Cramer.

1. On vous propose de montrer théoriquement que le processus de comptage

$$N_t = \sum_{k=1}^{\infty} \mathbb{I}_{T_k \leq t}, \quad T_k = \sum_{i=1}^{k} \tau_i$$

suit la loi de Poisson de paramètre λt.

Pour cela vous suivez les étapes :

• v.a. τ_i suit la loi exponentielle

Vous montrez par récurrence que $T_k = \sum_{i=1}^{k} \tau_i$ suit la loi de $Gamma(k, \lambda)$ de fonction de densité

$$f_{T_k}(x) = \frac{\lambda}{(k-1)!}(\lambda x)^{k-1} e^{-\lambda x}\, \mathbb{I}_{x\geq 0}$$

en utilisant le théorème suivante :

Soit v.a. ξ_1 possède une fonction de densité $f_{\xi_1}(x) = \lambda e^{-\lambda x}\, \mathbb{I}_{x\geq 0}$
v.a. ξ_2 possède une fonction de densité $f_{\xi_2}(x) = \lambda e^{-\lambda x}\, \mathbb{I}_{x\geq 0}$ alors
v.a. $Z = \xi_1 + \xi_2$ possède une fonction de densité $f_Z(x) = \int_{-\infty}^{\infty} \xi_1(x-y)\xi_2(y)dy$

1. *Réponse* :

Soit $k \in \mathbb{N}^*$:

soit $\tau_i \rightsquigarrow$ Exponentielle(λ) $\forall i \in [\![1,k]\!]$ avec $\lambda > 0$

On pose $T_k = \sum_{i=1}^{k} \tau_i$ et $\mathcal{P}_k$: " $T_k \rightsquigarrow$ Gamma(k,λ) et $f_{T_k}(x) = \frac{\lambda}{(k-1)!}(\lambda x)^{k-1} e^{-\lambda x}\, \mathbb{1}_{x>0}$ "

→ Pour $k=1$: la propriété $\mathcal{P}_1$ est vraie (par définition de τ_1)

→ Pour $k=2$: On a $\tau_1 \rightsquigarrow$ Exponentielle(λ) et $\tau_2 \rightsquigarrow$ Exponentielle(λ) donc τ_1 et τ_2 admettent des fonctions de densités f_{τ_1} et f_{τ_2} respectivement, d'après le théorème admis, $f_{T_2} = f_{\tau_1+\tau_2}$ admet une fonction de densité donnée par :

$$\forall x \in \mathbb{R} : f_{T_2}(x) = \int_{\mathbb{R}} f_{\tau_1}(x-y)\, f_{\tau_2}(y)\, dy = \int_{\mathbb{R}} \lambda e^{-\lambda(x-y)}\, \mathbb{1}_{]0,+\infty[}(x-y) \cdot \lambda e^{-\lambda y}\, \mathbb{1}_{]0,+\infty[}(y)\, dy$$

$$= \lambda^2 e^{-\lambda x} \int_{\mathbb{R}} \mathbb{1}_{]0,+\infty[}(x-y) \times \mathbb{1}_{]0,+\infty[}(y)\, dy$$

On fixe $y>0$: on a $x-y>0 \Rightarrow x>y \Rightarrow x>y>0$ et $x>0$

donc $\mathbb{1}_{]0,+\infty[}(x-y) \times \mathbb{1}_{]0,+\infty[}(y) = \mathbb{1}_{]0,x[}(y) \times \mathbb{1}_{]0,+\infty[}(x)$

Ainsi $$f_{T_2}(x) = \lambda^2 e^{-\lambda x} \cdot \mathbb{1}_{]0,+\infty[}(x) \cdot \int_{\mathbb{R}} \mathbb{1}_{]0,x[}(y)\, dy = \lambda^2 e^{-\lambda x} \mathbb{1}_{]0,+\infty[}(x) \int_0^x dy$$

Par suite $$f_{T_2}(x) = \frac{\lambda}{(2-1)!} (\lambda x)^{2-1} \cdot e^{-\lambda x} \cdot \mathbb{1}_{x>0}$$

Alors la propriété P_2 est vraie.

→ On suppose que la propriété P_k est vraie pour $k \in \mathbb{N}^*$ fixé et montrons que P_{k+1} est vraie :

Puisque $T_{k+1} = T_k + \tau_{k+1}$ et T_k, τ_{k+1} admettent des fonctions de densités alors d'après le théorème admis T_{k+1} admet une fonction de densité donnée par :

$$\forall x \in \mathbb{R} : \quad f_{T_{k+1}}(x) = \int_{\mathbb{R}} f_{T_k}(x-y) \cdot f_{\tau_{k+1}}(y)\, dy$$

Donc $$f_{T_{k+1}}(x) = \int_{\mathbb{R}} \frac{\lambda}{(k-1)!} (\lambda(x-y))^{k-1} e^{-\lambda(x-y)} \mathbb{1}_{]0,+\infty[}(x-y) \cdot \lambda e^{-\lambda y} \mathbb{1}_{]0,+\infty[}(y)\, dy$$

$$= \frac{\lambda \cdot \lambda^k}{(k-1)!} \cdot e^{-\lambda x} \int_{\mathbb{R}} (x-y)^{k-1} \mathbb{1}_{]0,+\infty[}(x-y) \mathbb{1}_{]0,+\infty[}(y)\, dy$$

On a déjà vu que $\mathbb{1}_{]0,+\infty[}(x-y) \times \mathbb{1}_{]0,+\infty[}(y) = \mathbb{1}_{]0,x[}(y) \times \mathbb{1}_{]0,+\infty[}(x)$

Donc $$f_{T_{k+1}}(x) = \frac{\lambda \cdot \lambda^k}{(k-1)!} \left(\int_{\mathbb{R}} (x-y)^{k-1} \mathbb{1}_{]0,x[}(y) \times \mathbb{1}_{]0,+\infty[}(x)\, dy \right) \cdot e^{-\lambda x}$$

$$= \frac{\lambda.\lambda^k}{(k-1)!}. e^{-\lambda x}\left(\int_0^x (x-y)^{k-1}\, dy\right) 1\!\!1_{]0,+\infty[}(x)$$

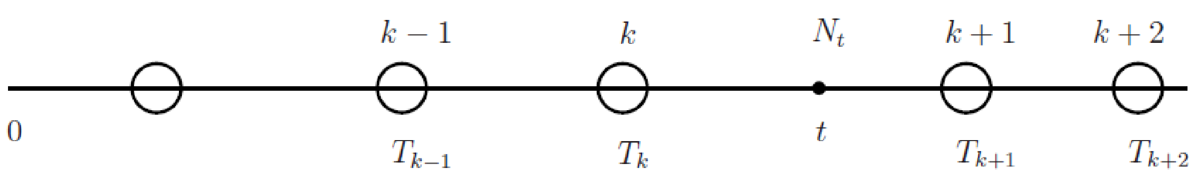
$$= \frac{\lambda.\lambda^k}{(k-1)!}\, e^{-\lambda x}\, 1\!\!1_{]0,+\infty[}(x) \cdot \left[-\frac{(x-y)^k}{k}\right]_0^x$$

Finalement. $f_{T_{k+1}}(x) = \frac{\lambda}{k!}(\lambda x)^{k+1-1} e^{-\lambda x}\, 1\!\!1_{x\geq 0}$ et P_{k+1} est vraie : Recurrence OK.

Conclusion : $\boxed{T_k = \sum_{i=1}^{k} \tau_i \text{ suit loi de Gamma } (k,\lambda) \text{ de fonction de densité } f_{T_k}(x) = \frac{\lambda}{(k-1)!}(\lambda x)^{k-1} e^{-\lambda x}\, 1\!\!1_{x\geq 0}}$

2. Soit t un instant fixe. Montrer que

$$\mathbb{P}[N_t = k] = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$$



Justifier d'abord que

$$\mathbb{P}[N_t = k] = \mathbb{P}[k \leq N_t < k+1] = \mathbb{P}[N_t \geq k] - \mathbb{P}[N_t \geq k+1]$$

Justifier aussi que

$$\mathbb{P}[N_t \geq k] = \mathbb{P}[T_k \leq t]$$

Vous continuez et utilisez l'expression de la fonction de répartition de v.a. T_k et T_{k+1}.

2. *Réponse* :

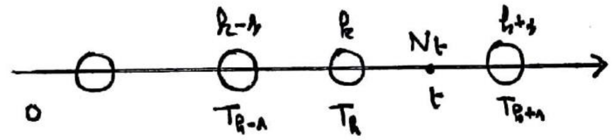
Par définition : $N_t = 1\!\!1_{\{\tau_1 \leq t\}} + 1\!\!1_{\{\tau_1+\tau_2 \leq t\}} + \cdots = 1 + 1 + 0 + 0 + \cdots \in \mathbb{N}$

Donc N_t est une v.a.r à valeur dans $\mathbb{N}$ et donc $\{k \leq N_t < k+1\} = \{N_t = k\}$ $\forall k \in \mathbb{N}$

D'autre part $\{N_t = k\} = \{N_t \geq k\} \setminus \{N_t \geq k+1\} \Rightarrow \mathbb{P}(N_t = k) = \mathbb{P}(N_t \geq k) - \mathbb{P}(N_t \geq k+1)$

Par suite : $\boxed{\mathbb{P}(N_t = k) = \mathbb{P}(k \leq N_t < k+1) = \mathbb{P}(N_t \geq k) - \mathbb{P}(N_t \geq k+1)}$

D'après le schéma :



Donc. $\{N_t \geq k\} = \{T_k \leq k\}$ et donc : $\boxed{\mathbb{P}(N_t \geq k) = \mathbb{P}(T_k \leq t)}$

Or d'après 1) on a $T_k \rightsquigarrow Gamma(k, \lambda)$ et $T_{k+1} \rightsquigarrow Gamma(k+1, \lambda)$

soit F_{T_k} et $F_{T_{k+1}}$ la fonction de répartition de T_k et T_{k+1} respectivement on a :

$$\mathbb{P}(N_t = k) = \mathbb{P}(N_t \geq k) - \mathbb{P}(N_t \geq k+1) = \mathbb{P}(T_k \leq t) - \mathbb{P}(T_{k+1} \leq t) = F_{T_k}(t) - F_{T_{k+1}}(t)$$

$$= \int_{-\infty}^{t} \frac{\lambda}{(k-1)!} (\lambda x)^{k-1} e^{-\lambda x} \mathbb{1}_{x>0} \, dx - \int_{-\infty}^{t} \frac{\lambda}{k!} (\lambda x)^{k} e^{-\lambda x} . \mathbb{1}_{x>0} . dx$$

$$= \int_{\mathbb{R}} \frac{\lambda}{(k-1)!} (\lambda x)^{k-1} e^{-\lambda x} \mathbb{1}_{]0,+\infty[\cap]-\infty,t]}(x) . dx - \int_{\mathbb{R}} \frac{\lambda}{k!} (\lambda x)^{k} e^{-\lambda x} \mathbb{1}_{]0,+\infty[\cap]-\infty,t]}(x) . dx$$

→ si $t \leq 0$: on a $]0,+\infty[\cap]-\infty,t] = \emptyset$ donc $\mathbb{P}(N_t = k) = 0$

→ si $t > 0$: on a $]0,+\infty[\cap]-\infty,t] =]0,t]$ donc $\mathbb{1}_{]0,+\infty[\cap]-\infty,t]}(x) = \mathbb{1}_{]0,t]}(x) . \mathbb{1}_{]0,+\infty[}(t)$

Donc : $$\mathbb{P}(N_t = k) = \int_{\mathbb{R}} \frac{\lambda}{(k-1)!} (\lambda x)^{k-1} e^{-\lambda x} \mathbb{1}_{]0,t]}(x) \, dx - \int_{\mathbb{R}} \frac{\lambda}{k!} (\lambda x)^{k} e^{-\lambda x} \mathbb{1}_{]0,t]}(x) \, dx$$

$$= \int_{0}^{t} \frac{\lambda}{(k-1)!} (\lambda x)^{k-1} e^{-\lambda x} dx . \mathbb{1}_{t>0} - \int_{0}^{t} \frac{\lambda}{k!} (\lambda x)^{k} e^{-\lambda x} dx . \mathbb{1}_{t>0}$$

$$= \mathbb{1}_{t>0} \left(\int_{0}^{t} \frac{\lambda}{(k-1)!} (\lambda x)^{k-1} e^{-\lambda x} dx - \int_{0}^{t} \frac{\lambda}{k!} (\lambda x)^{k} e^{-\lambda x} dx \right)$$

Par un changement de variable : $x = ut$ on a :

$$\mathbb{P}(N_t = k) = \mathbb{1}_{t>0} \cdot \left(\int_0^1 \frac{(\lambda t)^k}{(k-1)!} \cdot u^{k-1} \cdot e^{-\lambda t \cdot u} du - \int_0^1 \frac{(\lambda t)^{k+1}}{k!} u^k e^{-\lambda t \cdot u} du \right)$$

$$= \frac{(\lambda t)^k}{k!} \cdot \mathbb{1}_{t>0} \left(\int_0^1 k u^{k-1} e^{-\lambda t u} du - \int_0^1 \lambda t \cdot u^k e^{-\lambda t \cdot u} du \right)$$

Par une intégration par partie de fonctions de classe C^1 sur $[0,1]$:

$u \mapsto u^k$ et $u \mapsto \lambda t e^{-\lambda t u}$ on obtient :

$$\int_0^1 \lambda t \cdot u^k e^{-\lambda t u} du = \left[-u^k e^{-\lambda t u} \right]_0^1 + \int_0^1 k \cdot u^{k-1} e^{-\lambda t u} du$$

$$= -e^{-\lambda t} + \int_0^1 k \cdot u^{k-1} e^{-\lambda t u} du$$

Donc $\mathbb{P}(N_t = k) = \frac{(\lambda t)^k}{k!} \mathbb{1}_{t>0} \left(\left(\cancel{\int_0^1 k u^{k-1} e^{-\lambda t u} du} - \left(-e^{-\lambda t} + \cancel{\int_0^1 k \cdot u^{k-1} e^{-\lambda t u} du} \right) \right) \right)$

$$= \frac{(\lambda t)^k}{k!} e^{-\lambda t} \mathbb{1}_{t>0}$$

Par suite : Pour t un instant fixe : et $k \in \mathbb{N}$

$$\boxed{\mathbb{P}(N_t = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}}$$

3. Calculer l'espérance :

$$\mathbb{E}[X(t)], \quad X(t) = \sum_{k=1}^{N_t} Z_k.$$

Utilizer l'espérance conditionnelle :

$$\mathbb{E}[X(t)] = \sum_{k=0}^{\infty} \mathbb{E}[\sum_{i=1}^{k} Z_i | N_t = k] \cdot \mathbb{P}[N_t = k]$$

3. *Réponse* :

• Soit $X_{(t)} = \sum_{k=1}^{N_t} Z_k$

Fixons t et montrons que $X_{(t)}$ est intégrable :

on a $\mathbb{E}[|X_{(t)}|] = \sum_{m=1}^{+\infty} \int \mathbb{1}_{\{N_t=m\}} \left| \sum_{k=1}^{m} Z_k \right| dP \leq \sum_{m=1}^{+\infty} \mathbb{E}\left[\mathbb{1}_{\{N_t=m\}} \sum_{k=1}^{m} |Z_k|\right]$

or N_t et Z_k sont indépendantes donc aussi : $\mathbb{1}_{\{N_t=m\}}$ et $\sum_{k=1}^{m} |Z_k|$ sont indépendantes

d'où :

$$\mathbb{E}[|X_{(t)}|] \leq \sum_{m=1}^{+\infty} \mathbb{E}[\mathbb{1}_{\{N_t=m\}}] \times \mathbb{E}\left[\sum_{k=1}^{m} |Z_k|\right] = \sum_{m=1}^{+\infty} \mathbb{P}(N_t=m) \cdot \sum_{k=1}^{m} \mathbb{E}[|Z_k|]$$

Or les $(Z_k)_{k=1,m}$ sont indépendantes et identiquement distribuées de loi Gamma(α,β)
Alors $\mathbb{E}[|Z_k|] = \mathbb{E}[|Z_1|]$.

Donc : $\mathbb{E}[|X_{(t)}|] \leq \sum_{m=1}^{+\infty} \mathbb{P}(N_t=m)\left(m \cdot \mathbb{E}[|Z_1|]\right) = \mathbb{E}[|Z_1|] \cdot \sum_{m=1}^{+\infty} m \cdot \mathbb{P}(N_t=m)$

$$\leq \mathbb{E}[|Z_1|] \cdot \mathbb{E}[N_t] \quad \text{avec } \sum_{m=1}^{+\infty} m\, \mathbb{P}(N_t=m) = \sum_{m=1}^{+\infty} m \cdot \frac{(\lambda t)^m}{m!} = \lambda t$$

Ainsi $\mathbb{E}[|X_{(t)}|] \leq \lambda t \cdot \mathbb{E}[|Z_1|] < +\infty$ car $Z_1 \sim$ Gamma(α,β).

Par suite $X_{(t)}$ est intégrable et on a : en utilisant l'esperance conditionnelle :

$$\mathbb{E}[X_{(t)}] = \sum_{k=0}^{+\infty} \mathbb{E}\left[\sum_{i=1}^{N_t} Z_i \,\middle|\, \{N_t=k\}\right] \cdot \mathbb{P}(N_t=k)$$

Or $(Z_k)_{k=1,m}$ et N_t sont indépendantes donc $\mathbb{E}\left[\sum_{i=1}^{N_t} Z_i \,\middle|\, \{N_t=k\}\right] = \mathbb{E}\left[\sum_{i=1}^{k} Z_i\right]$

et donc $\mathbb{E}[X_{(t)}] = \sum_{k=0}^{+\infty} \mathbb{E}\left[\sum_{i=1}^{k} Z_i\right] \cdot \mathbb{P}(N_t=k) = \sum_{k=0}^{+\infty} \sum_{i=1}^{k} \mathbb{E}[Z_i] \cdot \mathbb{P}(N_t=k)$

puisque les $(Z_k)_{k=1,m}$ sont independantes et identiquement distribuées de loi Gamma(α,β) donc $\mathbb{E}[Z_i] = \mathbb{E}[Z_1]$ et alors :

$$\mathbb{E}[X_{(t)}] = \sum_{k=0}^{+\infty} k \cdot \mathbb{E}[Z_1] \cdot \mathbb{P}(N_t=k) = \mathbb{E}[Z_1] \cdot \sum_{k=0}^{+\infty} k \frac{(\lambda t)^k}{k!} = \lambda t \cdot \mathbb{E}[Z_1]$$

Donc : $\boxed{\mathbb{E}[X_{(t)}] = \lambda \cdot t \cdot \mathbb{E}[Z_1]}$ avec $\mathbb{E}[Z_1] = \frac{\alpha}{\beta}$

4. Calculer une espérance suivante très utile pour la suite :

$$\mathbb{E}[e^{\theta \cdot X(t)}], \quad \theta \in R^+$$

4. Réponse :

Calculons une espérance très utile : $\forall \theta \in \mathbb{R}^+ : \mathbb{E}[e^{\theta . X(t)}]$

C'est le calcul de la fonction génératrice des moments : $M_{X(t)}(\theta) = \mathbb{E}[e^{\theta . X(t)}]$

On a pour $\theta \in \mathbb{R}^+$ fixé : Par espérance conditionnelle :

$$M_{X(t)}(\theta) = \mathbb{E}[e^{\theta . X(t)}] = \sum_{k=0}^{+\infty} \mathbb{E}[e^{\theta . X(t)} | \{N_t = k\}] \cdot \mathbb{P}(N_t = k)$$

$$= \sum_{k=0}^{+\infty} \mathbb{E}[e^{\theta . \sum_{i=1}^{N_t} Z_i} | \{N_t = k\}] \cdot \mathbb{P}(N_t = k)$$

Par indépendance du processus $(N_t)_{t\geq 0}$ et la suite de v.a. (Z_k) on a :

$$\mathbb{E}[e^{\theta . \sum_{i=1}^{N_t} Z_i} | \{N_t = k\}] = \mathbb{E}[e^{\theta . \sum_{i=1}^{k} Z_i}]$$

Donc :

$$M_{X(t)}(\theta) = \sum_{k=0}^{+\infty} \mathbb{E}[e^{\theta . \sum_{i=1}^{k} Z_i}] \cdot \mathbb{P}(N_t = k)$$

Par indépendances des Z_i on a $\mathbb{E}[e^{\theta . \sum_{i=1}^{k} Z_i}] = \mathbb{E}[\prod_{i=1}^{k} e^{\theta . Z_i}] = \prod_{i=1}^{k} \mathbb{E}[e^{\theta . Z_i}]$

or $\mathbb{E}[e^{\theta . Z_i}]$ n'est que $M_{Z_i}(\theta)$ la fonction génératrice des moment de Z_i

or les Z_i sont indépendantes et identiquement distribuées de loi Gamma (α, β)

donc $M_{Z_i}(\theta) = M_{Z_1}(\theta)$ et donc $\mathbb{E}[e^{\theta . \sum_{i=1}^{k} Z_i}] = \prod_{i=1}^{k} M_{Z_1}(\theta) = (M_{Z_1}(\theta))^k$

Par suite :

$$M_{X(t)}(\theta) = \sum_{k=0}^{+\infty} (M_{Z_1}(\theta))^k \cdot \mathbb{P}(N_t = k) = G_{N_t}(M_{Z_1}(\theta))$$

avec G_{N_t} est la fonction génératrice de N_t donné par $G_{N_t}(u) = \mathbb{E}[u^{N_t}]$

Finalement : $\boxed{\forall \theta \in \mathbb{R}^+ : M_{X(t)}(\theta) = \mathbb{E}[e^{\theta . X(t)}] = G_{N_t}(M_{Z_1}(\theta))}$

Par calcul on obtient : $\left[\forall \theta \in \mathbb{R}^+ : M_{X(t)}(\theta) = e^{\lambda t (M_{Z_1}(\theta) - 1)}\right]$ avec $M_{Z_1}(\theta) = \left(\frac{\beta}{\beta - \theta}\right)^\alpha$

Livrable 1. - Activité 2. Simulations de Monte-Carlo

Simuler par Monte-Carlo en (Scilab ou Matlab ou Python) :

1. Processus de comptage N_t par deux façons différentes : à partir de la définition

$$N_t = \sum_{k=1}^{\infty} \mathbb{I}_{T_k \leq t} = \mathbb{I}_{\tau_1 \leq t} + \mathbb{I}_{\tau_1+\tau_2 \leq t} + \mathbb{I}_{\tau_1+\tau_2+\tau_3 \leq t} + \mathbb{I}_{\tau_1+\tau_2+\tau_3+\tau_4 \leq t} + \ldots$$

pour $t = 1$, $\lambda = 2, \lambda = 5$ et $\lambda = \frac{1}{24}$
Tracer les graphes de fonction de densité de Poisson.

1. *Réponse* : (Scilab et Python)

• **Méthode 1 :** en utilisant la définition de $N_t = \sum_{k=1}^{+\infty} \mathbb{1}_{\{T_k \leq t\}}$ avec $T_k = \sum_{i=1}^{k} \tau_i$

On sait que $\tau_i \sim$ Exponentielle(λ) avec $\lambda > 0$ $\forall i \in [\![1, k]\!]$, $k \geq 1$

Commençons d'abord de simuler τ_i :

→ Pour simuler une loi exponentielle on utilise la méthode :
inverser la fonction de répartition : $\tau_i \sim$ Exponentielle$(\lambda) \Rightarrow F_{\tau_i}(x) = (1 - e^{-\lambda x}) \mathbb{1}_{x>0}$

Donc : $F_{\tau_i}^{-1}(U) \sim$ Exponentielle(λ) avec $U \sim \mathcal{U}([0,1])$ et $F_{\tau_i}^{-1}(x) = -\frac{1}{\lambda} \ln(1-x)$

• fonction $[\tau_i]$ = V_A_Exponentielle(λ)
∘ Générer U = rand()
∘ Set $\tau_i = -\frac{1}{\lambda} \ln(1-U)$
• endfunction

→ Simulation de N_t :

```
• function [Nt] = V-A-Processus-Comptage(λ,t,N)
  ∘ Nt = 0
  ∘ T = 0                    # ici N remplace +∞ (N >> 1000)
  ∘ for k=1:N
        ∘ for i=1:k
        ∘ τ(i) = V-A-Exponentielle(λ)
        ∘ T = T + τ(i)
        ∘ endfor
  ∘ if  T ≤ t
  ∘ N(t) = N(t) + 1
  ∘ endif
  ∘ end for
• end function
```

→ Simulation d'une chaine de N_t :

```
• function [Nt] = Chaine-V-A-Processus-Comptage(λ,t,N)
  ∘ for m=1:Nmc
  ∘ Nt(m) = V-A-Processus-Comptage(λ,t,N)
  ∘ end for
• end function
```

On peut verifier la simulation grâce a' l'espérence empirique et la variance empirique

- **Methode 2 :** en utilisant la définition de $\mathbb{P}(N_t = k)$

→ **Simulation de N_t :**

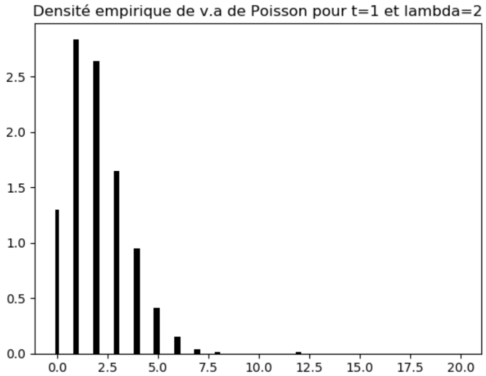
- fonction $[N_t]$ = V_A_Poisson_Composee(λ, t)
  - Set $n = 0$, proba $= e^{-\lambda t}$, $F =$ proba
  - Génér $U =$ rand()
  - While $U > F$
  - Set proba $= \frac{\lambda t}{n+1} \cdot$ proba, $F = F +$ proba, $n = n + 1$
  - endWhile
  - Set $X = n$
  - end function

→ **Simulation d'une chaine de N_t :**

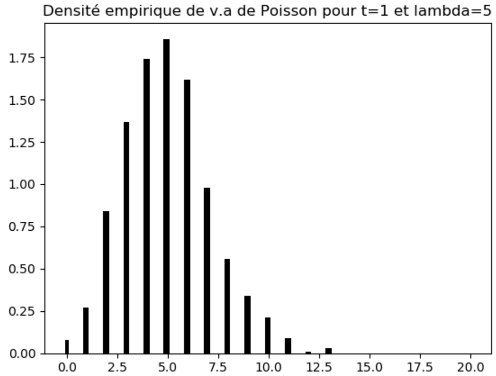
- function $[N_t]$ = Chaine_V_A_Poisson_Composee(λ, t)
  - for $n = 1 : N_{mc}$
  - $N_t(n)$ = V_A_Poisson_Composee(λ, t)
  - end for
  - end function

➔Les graphes de fonction de densité de Poisson : Python

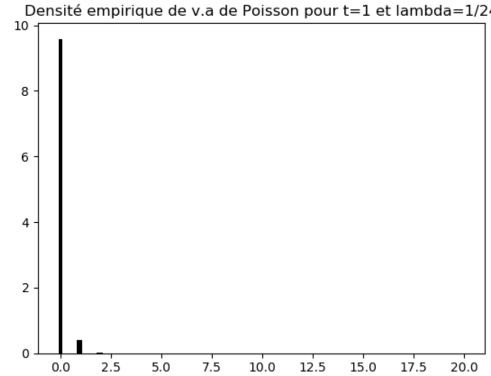
Densité empirique de v.a de Poisson pour t=1 et lambda=2



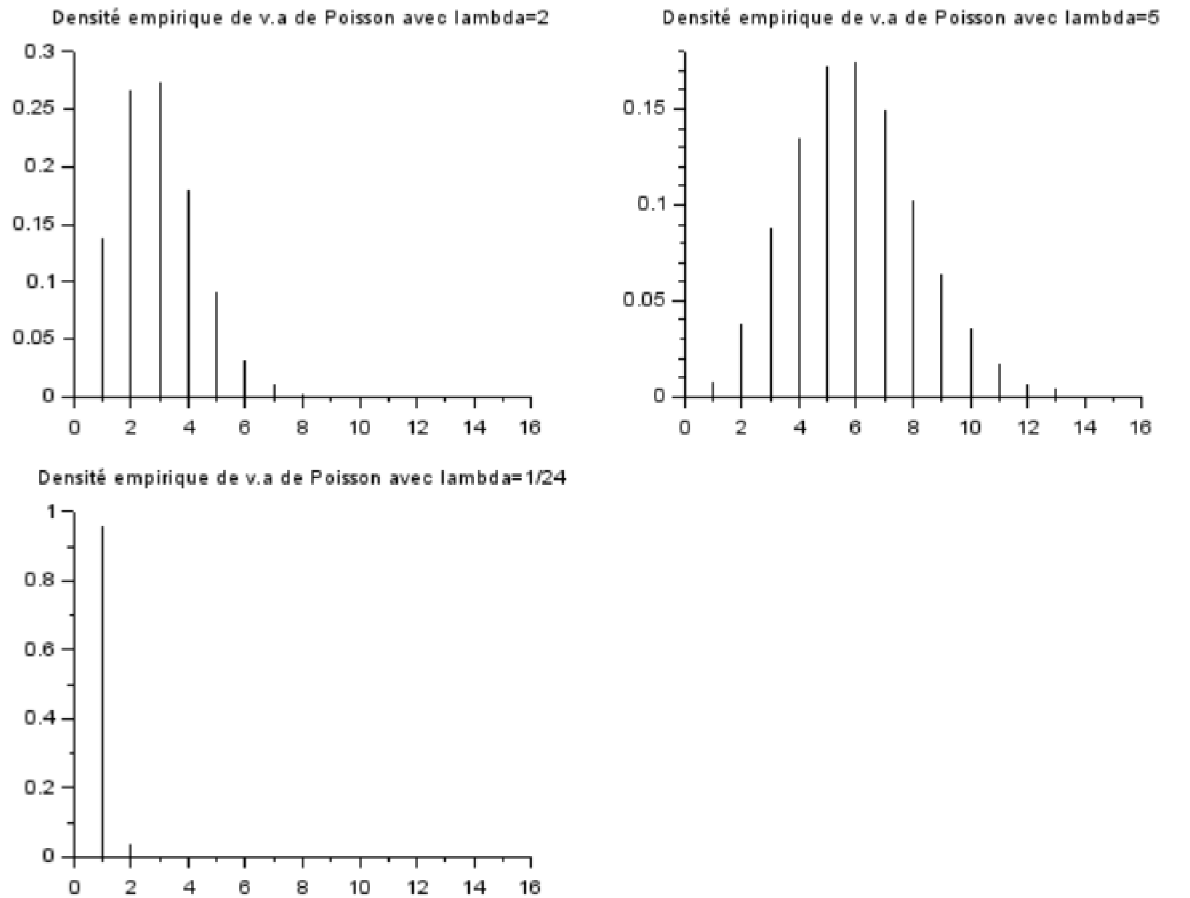
Densité empirique de v.a de Poisson pour t=1 et lambda=5



Densité empirique de v.a de Poisson pour t=1 et lambda=1/24



➔Les graphes de fonction de densité de Poisson : Scilab



2. Variable aléatoire Gamma(α, β) pour les deux cas : α est un nombre entier et α est quelconque.

• Dans le cas $\alpha = n$ simulez n fois une v.a. Y_i qui suivent la loi exponentielle de paramètre β. La variable aléatoire

$$X = Y_1 + Y_2 + Y_3 + \dots + Y_n$$

suit la loi de Gamma (n, β).

• Dans le deuxième cas vous utilisez la méthode de Rejet (voir le cours MCMC).

• Vous tracez les graphes de fonction de densité de Gamma pour les différentes paramètres : Gamma$(1, \frac{1}{2})$,Gamma$(2, \frac{1}{2})$, Gamma$(3, \frac{1}{2})$, Gamma$(5, 1)$, Gamma$(9, 2)$ Gamma$(2.5, \frac{1}{2})$ Gamma$(3.5, \frac{1}{2})$et comparer les graphes avec ceux de Wikipedia.

2. *<u>Réponse</u>* : (Scilab)

Soit $(Y_i)_{i=1,n}$ des v.a.r indépendantes de loi exponentielle (β) avec $\beta > 0$

On a $X = Y_1 + Y_2 + \dots + Y_n$ suit la loi Gamma (α, β) avec $\alpha = n$.

→ Simulation de loi Gamma $(\alpha = n, \beta)$: $(\alpha \in \mathbb{N}^*)$

```
• function [X] = V_A_Gamma(n,β)
o Set Y = V_A_Exponentielle(β)
o i=1 , U=rand()
o While i<n
o Y = Y - (1/β).ln(1-U)
o i=i+1
o endwhile
o Set X=Y
o end function
```

→ Simulation de loi Gamma (α, β) par la méthode de Rejet : $(\alpha, \beta \in \mathbb{R}_*^+)$

Soit $X \rightsquigarrow$ Gamma (α, β) on a $f_X(x) = \frac{1}{\Gamma(\alpha)} \cdot \beta^\alpha x^{\alpha-1} e^{-\beta x} \mathbb{1}_{x>0}$ sa fonction de densité.

si $\alpha \notin \mathbb{N}$ on utilise la méthode de rejet : soit $Y \rightsquigarrow$ Gamma (n, γ) qu'on sait déja simuler. On a $C = \sup_{x>0} \frac{f_X(x)}{g_Y(x)}$ avec $g_Y(x) = \frac{1}{\Gamma(n)} \gamma^n x^{n-1} e^{-\gamma x} \mathbb{1}_{x>0}$ la densité de Y.

Alors :

$$C = \sup_{x>0} \frac{f_X(x)}{g_Y(x)} = \sup_{x>0} \frac{\frac{1}{\Gamma(\alpha)} \cdot \beta^\alpha \cdot x^{\alpha-1} \cdot e^{-x\beta}}{\frac{1}{\Gamma(n)} \cdot \gamma^n x^{n-1} \cdot e^{-x\gamma}} = \sup_{x>0} \frac{\beta^\alpha}{\gamma^n} \cdot \frac{\Gamma(n)}{\Gamma(\alpha)} \cdot x^{\alpha-n} \cdot e^{-x(\beta-\gamma)}$$

Après avoir étudier la fonction $x \longmapsto x^{\alpha-n} e^{-x(\beta-\gamma)}$ on trouve le sup : $\left[x = \frac{\alpha-n}{\beta-\gamma}\right]$

donc :

$$\boxed{C = \frac{\Gamma(n)}{\Gamma(\alpha)} \cdot \frac{\beta^\alpha}{\gamma^n} \cdot \left(\frac{\alpha-n}{\beta-\gamma}\right)^{\alpha-n} \cdot e^{-(\alpha-n)}}$$

pour avoir $C \geq 1$ on choisit : $\beta > \gamma$ et $n < \alpha < n+1$

D'après le théorème de Rejet : On peut simuler X.

- function [f] = f(x, α, β)
- $f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$
- end function

- function [g] = g(x, m, γ)
- $g(x) = \frac{\gamma^m}{\Gamma(m)} . x^{m-1} e^{-\gamma x}$
- end function

- function [X] = Rejet_Gamma ( )
- for m = 1 : Nmc
- Générer U = rand()
- Y = V_A_Gamm(m, γ)
- if $U \leq \frac{f_X(Y)}{C . g_Y(Y)}$
- Set X(m) = Y
- endif
- end for
- end function

- Alors pour : $\alpha = 2{,}5$ et $\beta = \frac{1}{2}$

  on choisit : $m = 2$ et $\gamma = \frac{1}{4}$

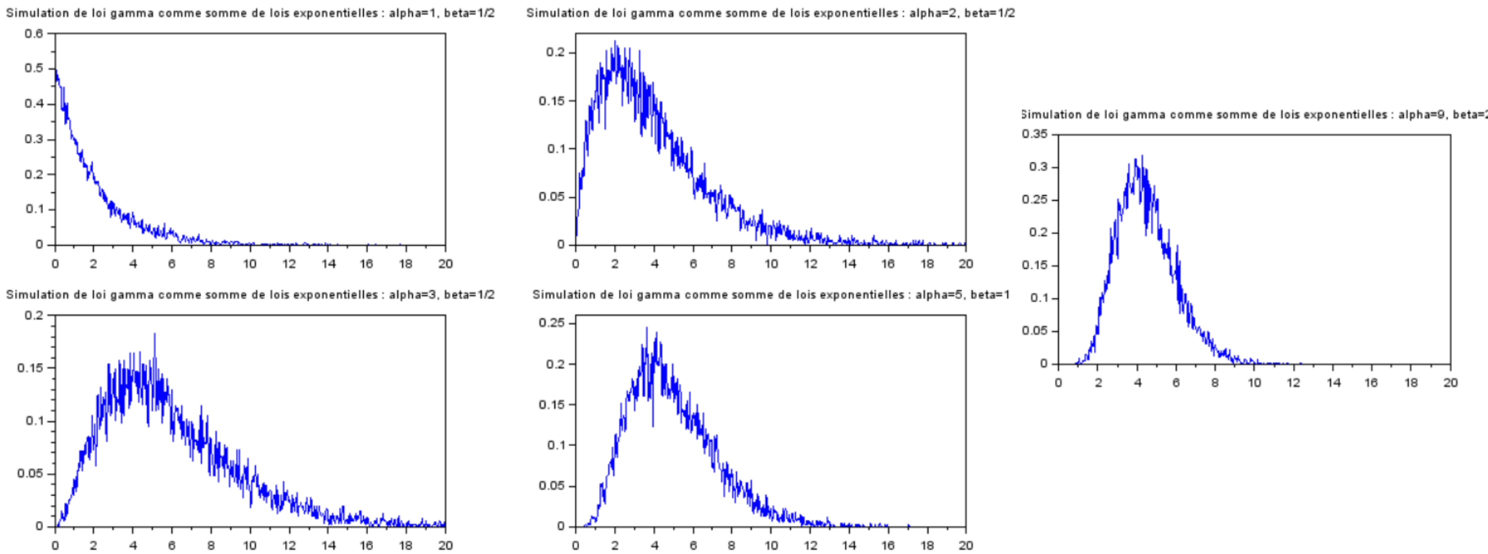
- Pour : $\alpha = 3{,}5$ et $\beta = \frac{1}{2}$

  on choisit : $m = 3$ et $\beta = \frac{1}{4}$

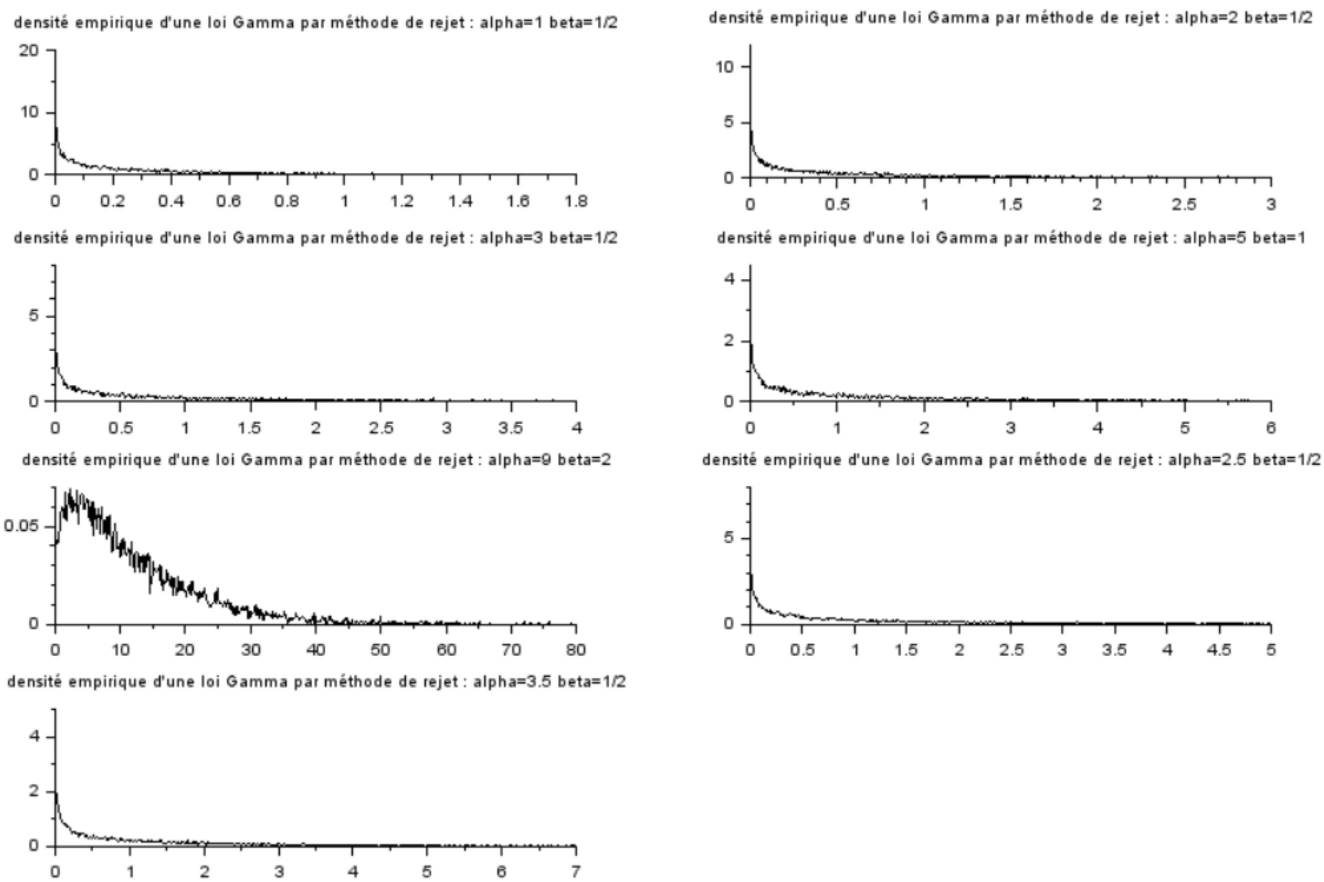
On peut aussi utiliser comme fonction g la densité d'une loi de Weibull (x>0)

$$f(x;k,\lambda) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \mathrm{e}^{-(x/\lambda)^k}$$

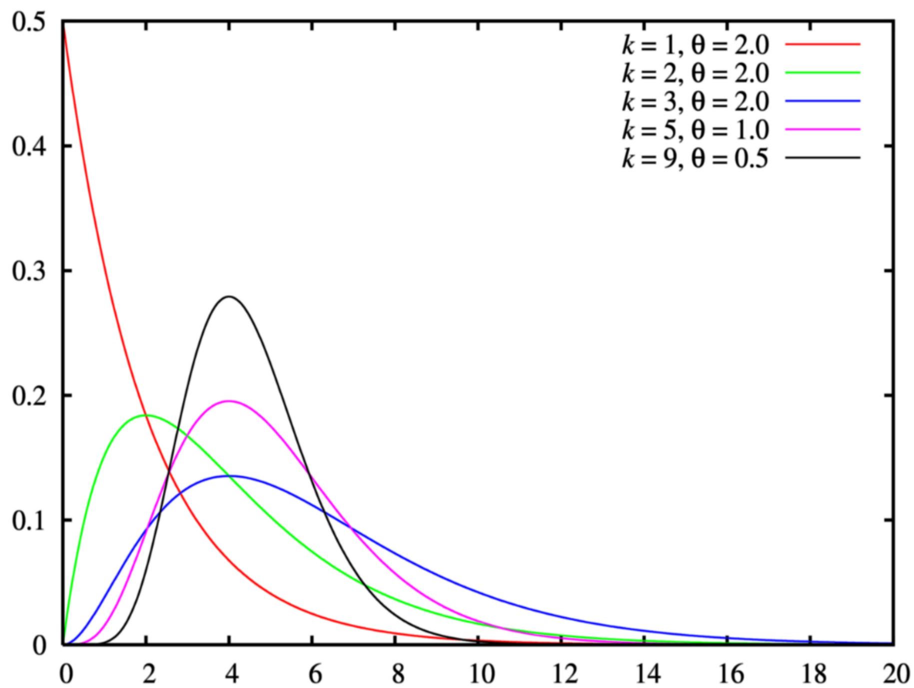
➔Les graphes de fonction de densité de Gamma (alpha entier) : Scilab



➔Les graphes de fonction de densité de Gamma (alpha >0) : Scilab



➔Comparaison avec les graphes de Wikipédia :



On remarque que les densités empiriques simulées pour la loi Gamma ressemblent beaucoup au graphe de Wikipédia, alors on peut dire qu'on a bien réussi à simuler la loi de Gamma.

2. Variable aléatoire Pareto. V. a. Pareto modélise des dommages causés par Covid -19. Choisissez des jeux de données et tracez les fonction de densités.

3. *Réponse* : (Scilab et Python)

Le principe de Pareto, aussi appelé loi de Pareto, est un phénomène empirique constaté dans certains domaines : environ 80 % des effets sont le produit de 20 % des causes.

Soit X une v.a de loi Pareto(a,b) avec $a>0$ et $b>0$

Sa fonction de densité est : $\forall x>b$: $f_X(x) = \frac{a \cdot b^a}{x^{a+1}} \mathbb{1}_{x>b}$ donc $F_X(x) = \left(1 - \left(\frac{b}{x}\right)^a\right) \mathbb{1}_{x>b}$

Alors $F_X^{-1}(x) = \frac{b}{(1-x)^{1/a}}$.

→ Simulation de loi Pareto(a,b) par inversion de F_X :

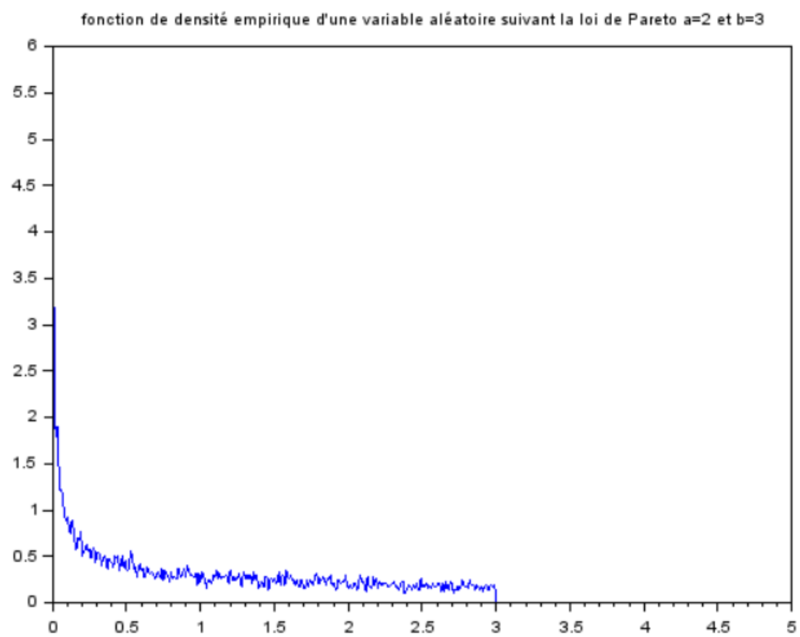
```
• function [X] = V-A-Pareto(a,b)
o Générer U = rand()
o Set X = b.(1-U)^(-1/a)
• endfunction
```

→ Simulation d'une chaine de Pareto(a,b) :

```
• function [X] = Chaine_V-A_Pareto(a,b)
o for n = 1:Nmc
o X(n) = V-A-Pareto(a,b)
o end for
• endfunction
```

Soit le paramètre a de loi de Pareto : le nombre de personnes qu'un malade infecte en moyenne autour de lui ne serait pas de 2 ou 2,5 comme le suggèrent les rapports chinois relayés par l'Organisation mondiale de la santé (OMS). Soit b = 3 le taux de mort. (a=2,b=3)

➔Les graphes de fonction de densité de Pareto : Scilab



➔Les graphes de fonction de densité de Pareto : Python