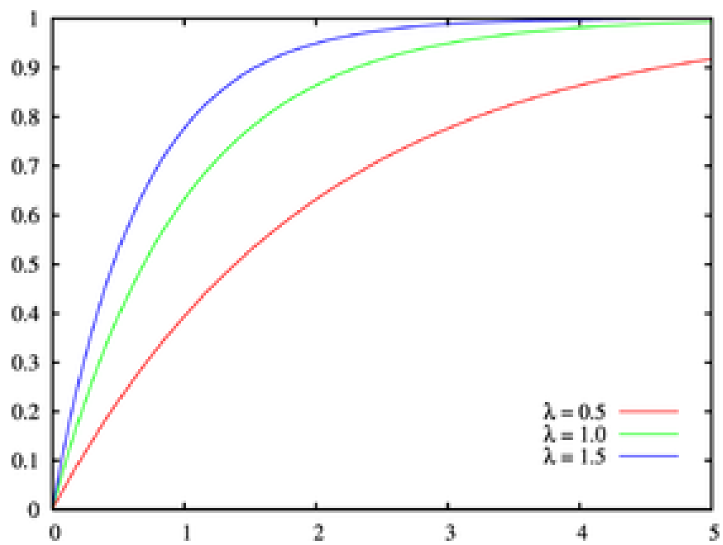
Institut national des sciences appliquées de Rouen

INSA de Rouen

Projet MSRO GM3 - Vague 2 - Sujet 1

Simulation d'une loi exponentielle de paramètre λ



Auteurs :
Thibaut André-Gallis
thibaut.andregallis@insa-rouen.fr
Kévin Gatel
kevin.gatel@insa- rouen.fr

Enseignant :
Ioana Ciotir
ioana.ciotir@insa-rouen.fr

27 Novembre 2020

Table des matières

Introduction

La loi exponentielle modélise la durée de vie d'un phénomène sans mémoire ou sans vieillissement. En d'autres termes, la probabilité que le phénomène dure au moins $h + t$ heures sachant qu'il a déjà duré t heures sera la même que la probabilité de durer h heures à partir de sa mise en fonction initiale. Elle peut notamment modéliser la vie des circuits électriques, résoudre des problématiques de durée de vie en général.

Dans ce projet, nous allons nous intéresser à comment simuler et modéliser une loi exponentielle de paramètre λ. Dans un premier temps, nous allons démontrer qu'il est possible d'obtenir une loi exponentielle $G(U)$ en partant d'une loi uniforme U, ensuite, nous allons démontrer que prendre une loi uniforme U ou bien $1 - U$ ne change rien pour en obtenir une. Dans un deuxième temps, nous allons vérifier expérimentalement qu'il est possible de générer une loi exponentielle à partir d'une loi uniforme et vérifier que $G(U) \sim G(1 - U)$.

1. Partie théorique

Dans cette partie, nous allons démontrer les deux premières questions posées.

1.1 G(U) $\sim \varepsilon(\lambda)$

Soit U une loi uniforme sur [0,1].
On a :

$$\mathbb{P}(U \leq t) = F_U(t) = \begin{cases} 0 & \text{si } t < 0 \\ \frac{t-0}{1-0} = t & \text{si } t \in [0,1] \\ 1 & \text{si } t > 1 \end{cases}$$

Soit $\lambda > 0$ et G une fonction telle que :

$$\begin{array}{rcccc} G & : & [0,1] & \longrightarrow & \mathbb{R}^+ \\ & & u & \longmapsto & \frac{-1}{\lambda}\ln(1-u) \end{array}$$

Alors :

$$\mathbb{P}(G(U) < t) = \mathbb{P}(\frac{-1}{\lambda}\ln(1-U) < t) = \mathbb{P}(\ln(1-U) > -\lambda t) = \mathbb{P}(1-U > e^{-\lambda t})$$

Or :

$$\mathbb{P}(1-U > e^{-\lambda t}) = \mathbb{P}(U < 1-e^{-\lambda t}) = F_U(1-e^{-\lambda t}) = \begin{cases} 0 & \text{si } 1-e^{-\lambda t} < 0 \\ 1-e^{-\lambda t} & \text{si } 1-e^{-\lambda t} \in [0,1] \\ 1 & \text{si } 1-e^{-\lambda t} > 1 \end{cases}$$

On a :

$$\begin{array}{rcl} 1-e^{-\lambda t} < 0 & \iff & e^{-\lambda t} > 1 \\ & \iff & \underbrace{-\lambda}_{<0} t > 0 \\ & \iff & t < 0 \end{array}$$

Ensuite :

$$\begin{array}{rcl} 1-e^{-\lambda t} \in [0,1] & \iff & 0 \leq 1-e^{-\lambda t} \leq 1 \\ & \iff & e^{-\lambda t} \leq 1 \\ & \iff & \underbrace{-\lambda}_{<0} t \leq 0 \\ & \iff & t \geq 0 \end{array}$$

Enfin :

$$\begin{array}{rcl} 1-e^{-\lambda t} > 1 & \iff & e^{-\lambda t} < 0 \\ & \iff & t \in \emptyset \end{array}$$

Ainsi :

$$\mathbb{P}(G(U) < t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 - e^{-\lambda t} & \text{si } t \geq 0 \end{cases}$$

D'où $G(U) \sim \varepsilon(\lambda)$.

1.2 G(U) $\sim$ G(1-U) $\sim \varepsilon(\lambda)$

On a :

$$\mathbb{P}(G(1-U) < t) = \mathbb{P}(\frac{-1}{\lambda}\ln(U) < t) = \mathbb{P}(\ln(U) > -\lambda t)$$

D'où :

$$\mathbb{P}(\ln(U) > -\lambda t) = \mathbb{P}(U > e^{-\lambda t}) = 1 - \mathbb{P}(U < e^{-\lambda t}) = 1 - F_U(e^{-\lambda t})$$

Or :

$$F_U(e^{-\lambda t}) = \begin{cases} 0 & \text{si } e^{-\lambda t} < 0 \text{ Aucun t ne satisfait l'inéquation} \\ e^{-\lambda t} & \text{si } e^{-\lambda t} \in [0,1] \\ 1 & \text{si } e^{-\lambda t} > 1 \end{cases}$$

Donc :

$$-F_U(e^{-\lambda t}) = \begin{cases} -e^{-\lambda t} & \text{si } -e^{-\lambda t} \in [-1,0] \\ -1 & \text{si } -e^{-\lambda t} < -1 \end{cases}$$

Enfin :

$$1 - F_U(e^{-\lambda t}) = \mathbb{P}(G(1-U) < t) = \begin{cases} 1 - e^{-\lambda t} & \text{si } 1 - e^{-\lambda t} \in [0,1] \iff t \geq 0 \\ 0 & \text{si } 1 - e^{-\lambda t} < 0 \iff t < 0 \end{cases} = F_U(1 - e^{-\lambda t})$$

On retombe bien sur la même loi exponentielle que $G(U)$.

Pour conclure si $U \sim \mathbb{U}([0,1])$ on aura $G(1-U) \sim G(U) \sim \varepsilon(\lambda)$.

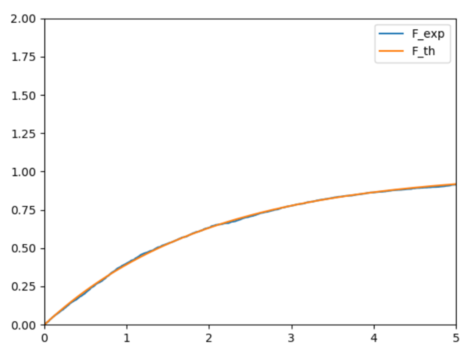
2. Partie appliquée

Dans cette partie, nous allons illustrer ce que nous venons de démontrer en construisant un exemple. Nous avons ainsi fait un programme qui, en partant d'une simple loi uniforme que possèdent les langages traditionnels, génère une loi exponentielle [1]. Nous allons ensuite analyser ces résultats afin de discuter de la méthode et de la conformité.

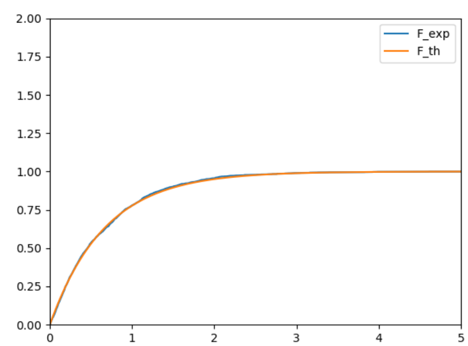
2.1 Construction et analyse de l'exemple

Dans un premier temps, nous avons tiré puis stocké un grand nombre de valeurs (ici N=2000) tirées aléatoirement de manière uniforme entre [0,1]. La fonction G est appliquée au tableau puis ce dernier est trié de manière à classer par ordre croissant les valeurs. Nous avons ensuite utilisé un second tableau compteur permettant de comptabiliser le nombre de valeur inférieur à une seconde valeur constante appartenant au tableau créé précédemment. Enfin, en incrémentant cette constante on peut ainsi construire une fonction de répartition de nos valeurs tirées initialement de manière aléatoire. On trace ensuite les deux courbes pour pouvoir les comparer. Pour pouvoir comparer notre résultat obtenu à la courbe théorique afin de valider notre simulation, nous avons tracé la fonction de répartition d'une loi exponentielle de paramètre λ nous servant ainsi de modèle.

On obtient ce résultat :



Fonction de répartition avec $\lambda = 0.5$



Fonction de répartition avec $\lambda = 1.5$

On obtient des courbes très proches voire superposées.

Pour avoir d'autres éléments à étudier, nous avons comparé l'espérance obtenue avec celle théorique.

1. Le code est disponible en annexe

Pour λ=0.5

$\mathbb{E}_{th}$	$\mathbb{E}_{exp}$
2	2.01158

Pour λ=1.5

$\mathbb{E}_{th}$	$\mathbb{E}_{exp}$
0.66667	0.67476

Les espérances sont également très proches. On calcule aussi numériquement l'écart moyen entre les deux courbes en plus du tracé de la courbe. On obtient ainsi :

Pour λ=0.5

Ecart
0.00963

Pour λ=1.5

Ecart
0.00557

Nous retrouvons bien toutes les propriétés d'une loi exponentielle. La loi exponentielle formée par $G(U)$ est donc bien vérifiée expérimentalement.

2.2 Cohérence de U et $1-U$ pour G

Pour se faire, nous avons fabriqué de la même manière la fonction de répartition en prenant $1-U$ au lieu de U.

Nous obtenons :

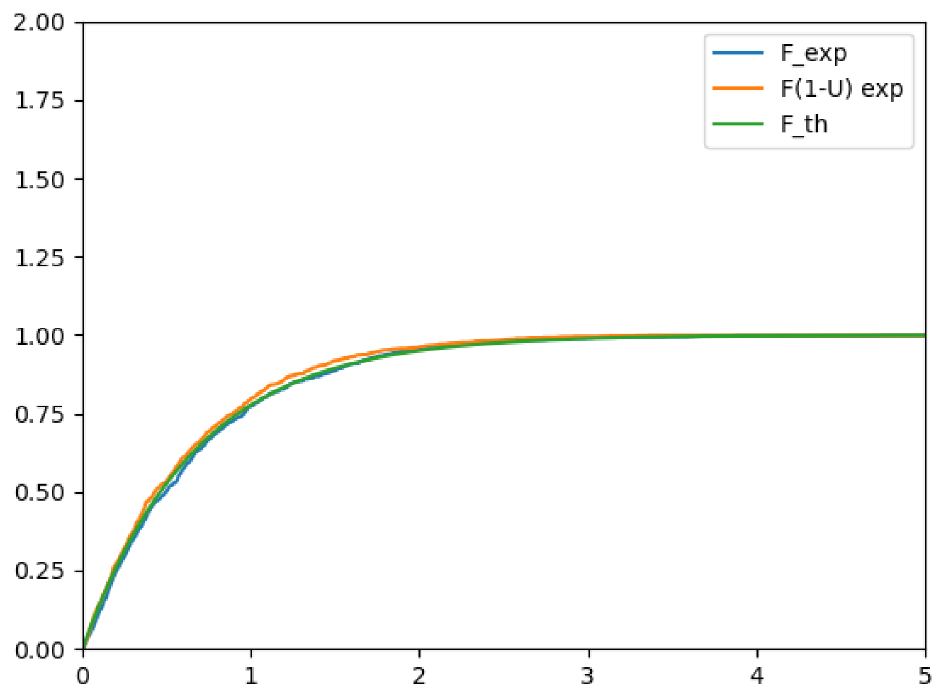


FIGURE 2.1 – Fonctions de répartition

Nous observons que les deux courbes ont la même allure et sont très proches. De plus, nous obtenons cette figure avec une moyenne des écarts entre les deux courbes expérimentales de l'ordre de 1%.

Ecart
0.01038

Ainsi, nous pouvons donc conclure sur la cohérence de U et $1-U$ pour la fonction G.

Conclusion

Pour conclure ce projet, rappelons le problème initial. Nous avons étudié la loi exponentielle de paramètre λ d'un point de vue mathématique puis informatique. Pour se faire, nous avons simulé une loi exponentielle à partir d'une loi uniforme. Ce projet a donc été réalisé en deux parties : une mathématique et une plus informatique.

En première partie, nous nous sommes intéressés à l'aspect purement mathématique. Nous disposions d'une certaine fonction G telle que :

$$\begin{array}{cccc} G & : & [0,1] & \longrightarrow & \mathbb{R}^+ \\ & & u & \longmapsto & \frac{-1}{\lambda}\ln(1-u) \end{array}$$

Et d'une variable aléatoire U suivant une loi uniforme sur [0,1]. Nous avons donc démontré mathématiquement que la variable $G(U)$ suivait une loi exponentielle de paramètre λ. Nous avons ensuite démontré que si l'on remplaçait U par $1-U$ dans G, on trouvait le même résultat.

Dans un second temps, nous avons essayé d'implémenter un programme informatique permettant d'illustrer cette démonstration à travers un exemple. Nous savions que la plupart des langages possèdent une fonction aléatoire qui suit une loi uniforme sur [0,1]. Notre programme consiste à faire de nombreux tirages aléatoires indépendants en nous servant de cette fonction et de la fonction G. Pour ensuite vérifier que ces valeurs suivaient bien une loi exponentielle de paramètre λ, nous avons eu l'idée de tracer la fonction de répartition de ces valeurs et de la comparer avec la fonction de répartition théorique. De plus, nous avons comparé l'espérance de nos mesures avec l'espérance théorique. Nous avons réitéré l'expérience avec la variable $1-U$ et constaté la même chose.

Nous avons eu des difficultés pour obtenir une courbe lisse avec un nombre fini de valeurs mais nous avons finalement trouvé un bon équilibre permettant d'obtenir une courbe de bonne qualité très proche de la courbe théorique sans surcharger l'utilisation de la mémoire et le nombre d'opérations qui rendait le programme très long à s'exécuter.

Pour finir, ce projet nous aura permis de nous familiariser avec un langage supplémentaire qu'est le python, ainsi que des bibliothèques comme *matplotlib.pyplot* très utile pour tracer des graphiques.

Nous avons trouvé très intéressant de pouvoir lier démonstrations mathématiques avec applications informatiques pour illustrer quelque chose de théorique qui peut paraître très abstrait avec un exemple plus concret.

C'est à partir de deux points de vues différents que nous avons pu obtenir un programme performant et précis qui donne des résultats en accord avec ceux attendus pour toute valeur de λ.

Il existe probablement d'autres façons d'obtenir des lois de probabilités diverses et variées à partir d'une loi uniforme sur [0,1]. On pourrait donc utiliser le même procédé pour modéliser des lois de probabilités sur ordinateur ce qui pourrait s'avérer utile pour faire des projections de l'avenir sur l'évolution d'une pandémie par exemple.

Annexe

Code 2.1 – Programme en Python

```
import numpy as np
import random
import matplotlib.pyplot as plt
import math as m

N=1000
Lambda=1.5
esperance=0.0
ecart_moyen=0.0
tab=np.zeros(N)
tabu=np.zeros(N)

for i in range(N):
    tab[i]=random.random()
    tabu[i]=1-tab[i]                              #construction de G(1-U)
    tab[i]=(-1/Lambda)*m.log(1-tab[i])
    tabu[i]=(-1/Lambda)*m.log(1-tabu[i])

tabtrie=sorted(tab)    #tableau trie
tabutrie=sorted(tabu)

t1 = np.arange(0.0,5.0,5.0/N)    #tableau N valeurs entre 0 5 par pas 5/N

tabcompteur = np.zeros(N)
tabucompteur = np.zeros(N)

for i in range(N):
    for j in range(N):
        if (tabtrie[j]<=t1[i]) or ((tabtrie[j]>=5.0)and(t1[i]>=5.0)):
            tabcompteur[i]+=1
        if (tabutrie[j]<=t1[i]) or ((tabutrie[j]>=5.0)and(t1[i]>=5.0)):
            tabucompteur[i]+=1
F=tabcompteur/N
FU=tabucompteur/N

ecartU=0                    #ecart moyen entre F et FU
for i in range(N):
        ecartU+=abs(F[i]-FU[i])
ecartU=ecartU/N

plt.plot(t1,F,label='F_exp')
plt.plot(t1,FU,label='F(1-U) exp')
```

```
F_th=np.zeros(N)
for i in range(N):
    F_th[i]=1-m.exp(-Lambda*tabtrie[i])

plt.plot(tabtrie,F_th,label='F_th')

plt.legend()
plt.axis([0, 5, 0, 2])
plt.show()

for i in range(N):
    ecart_moyen+=abs(F[i]-(1-m.exp(-Lambda*t1[i])))
ecart_moyen=ecart_moyen/N


for i in range(N):
    esperance+=tabtrie[i]
esperance=esperance/N

print()
print("Nous avons pris lambda = ",Lambda)
print("Notre esperance theorique est donc: 1/lambda = ",round(1/Lambda,5))

print("Notre esperance experimentale = ",esperance)

print("En calculant l'ecart moyen entre les deux courbes on trouve ")
print(round(ecart_moyen,5))
print("Donc notre courbe experimentale se superpose presque ")
print("parfaitement avec la courbe theorique")

print("Cette difference est toujours tres faible , et comfirme ")
print("que notre tirage se rapproche bien d'un tirage aleatoire ")
print("selon une loi exponentielle de parametre lambda = ")
print(round(Lambda,5))

print("L'ecart moyen entre G(U) et G(1-U) vaut ",round(ecartU,5))
```