

Gestion de flux dans le réseau

TD n ° 4

Modélisation mathématique

Q4

Sibylle Roux

Juliette Arazo

Nicolas Le Gallo

Tanguy Thomas

14 novembre 2017

## Table des matières

<b>I</b>	<b>Etude statistique des temps interarrivés</b>	<b>5</b>
<b>1</b>	<b>Etude statistique des temps interarrivés pour tous les serveurs</b>	<b>5</b>
1.1	Indicateurs de position et de dispersion . . . . .	5
1.2	Fonction de répartition . . . . .	5
1.3	Histogramme . . . . .	6
1.3.1	Histogramme avec classes isoamplitudes . . . . .	6
1.3.2	Histogramme avec classes isofréquences . . . . .	7
<b>II</b>	<b>Etude statistique des temps de service</b>	<b>7</b>
<b>2</b>	<b>Indicateurs de position et de dispersion</b>	<b>8</b>
<b>3</b>	<b>Fonctions de répartition</b>	<b>8</b>
3.1	Serveur 1 . . . . .	8
3.2	Serveur 2 . . . . .	9
3.3	Serveur 3 . . . . .	10
<b>4</b>	<b>Histogrammes</b>	<b>11</b>
4.1	Serveur 1 . . . . .	11
4.2	Serveur 2 . . . . .	12
4.3	Serveur 3 . . . . .	12
<b>III</b>	<b>Ajustement graphique à des lois mathématiques</b>	<b>13</b>
<b>5</b>	<b>Estimation des paramètres</b>	<b>13</b>
5.1	Loi normale . . . . .	13
5.2	Loi uniforme . . . . .	13
5.3	Loi exponentielle . . . . .	13
<b>6</b>	<b>Tous les serveurs</b>	<b>14</b>
6.1	Superposition des fonctions de répartition . . . . .	14
6.2	Superposition des fonctions de densité et de l'histogramme . . . .	14
6.3	Propriété d'absence de mémoire . . . . .	15
<b>7</b>	<b>Serveur 1</b>	<b>15</b>
7.1	Superposition des fonctions de répartition . . . . .	15
7.2	Superposition des fonctions de densité et de l'histogramme . . . .	16
7.3	Propriété d'absence de mémoire . . . . .	16

<b>8</b>	<b>Serveur 2</b>	<b>17</b>
8.1	Superposition des fonctions de répartition . . . . .	17
8.2	Superposition des fonctions de densité et de l'histogramme . . . .	18
8.3	Propriété d'absence de mémoire . . . . .	18
<b>9</b>	<b>Serveur 3</b>	<b>19</b>
9.1	Superposition des fonctions de répartition . . . . .	19
9.2	Superposition des fonctions de densité et de l'histogramme . . . .	19
9.3	Propriété d'absence de mémoire . . . . .	20
<b>IV</b>	<b>Conclusion</b>	<b>20</b>
<b>A</b>	<b>Etude statistique des temps interarrivés</b>	<b>22</b>
A.1	Indicateurs de position et de dispersion . . . . .	22
A.2	Fonction de répartition . . . . .	22
A.3	Histogramme . . . . .	22
A.3.1	Histogramme avec classes isoamplitudes . . . . .	22
A.3.2	Histogramme avec classes isofréquences . . . . .	23
<b>B</b>	<b>Etude statistique des temps de service</b>	<b>23</b>
B.1	Indicateurs de position et de dispersion . . . . .	23
B.1.1	Serveur 1 . . . . .	23
B.1.2	Serveur 2 . . . . .	24
B.1.3	Serveur 3 . . . . .	24
B.2	Fonctions de repartitions . . . . .	25
B.2.1	Serveur 1 . . . . .	25
B.2.2	Serveur 2 . . . . .	25
B.2.3	Serveur 3 . . . . .	25
B.3	Histogrammes . . . . .	26
B.3.1	Serveur 1 . . . . .	26
B.3.2	Serveur 2 . . . . .	26
B.3.3	Serveur 3 . . . . .	27
<b>C</b>	<b>Ajustement graphique à des lois mathématiques</b>	<b>27</b>
C.1	Tous les serveurs . . . . .	27
C.1.1	Superposition des fonctions de répartitions . . . . .	27
C.1.2	Superposition des fonctions de densité et de l'histogramme	28
C.1.3	Propriété d'absense de mémoire . . . . .	29
C.2	Serveur 1 . . . . .	30
C.2.1	Superposition des fonctions de répartitions . . . . .	30
C.2.2	Superposition des fonctions de densité et de l'histogramme	31
C.2.3	Propriété d'absense de mémoire . . . . .	32
C.3	Serveur 2 . . . . .	33
C.3.1	Superposition des fonctions de répartitions . . . . .	33
C.3.2	Superposition des fonctions de densité et de l'histogramme	34

	C.3.3	Propriété d'absence de mémoire . . . . .	35
C.4	Serveur 3	. . . . .	36
	C.4.1	Superposition des fonctions de répartitions . . . . .	36
	C.4.2	Superposition des fonctions de densité et de l'histogramme	37
	C.4.3	Propriété d'absence de mémoire . . . . .	37

## Première partie

# Etude statistique des temps interarrivés

## 1 Etude statistique des temps interarrivés pour tous les serveurs

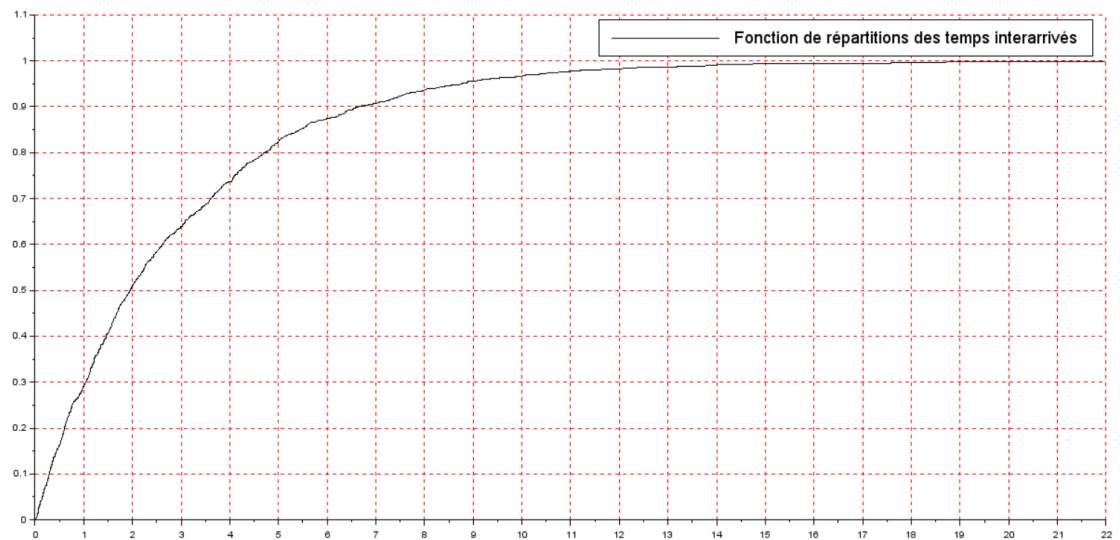
### 1.1 Indicateurs de position et de dispersion

Min	Max	Moyenne	Médiane
0.01	21.98	2.9	1.95

Variance	Ecart-type	Etendue
8.73	2.95	22

Q1	Q2	Interquartile
0.76	4.09	3.33

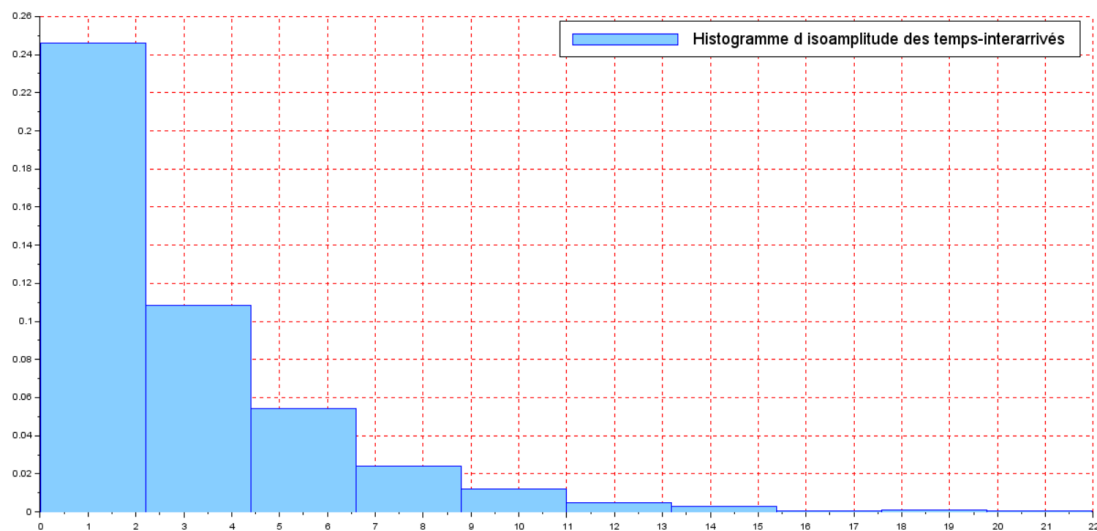
### 1.2 Fonction de répartition



Ce graphique nous montre que la moitié des temps inter-arrivées sont inférieurs ou égaux à 2 secondes.

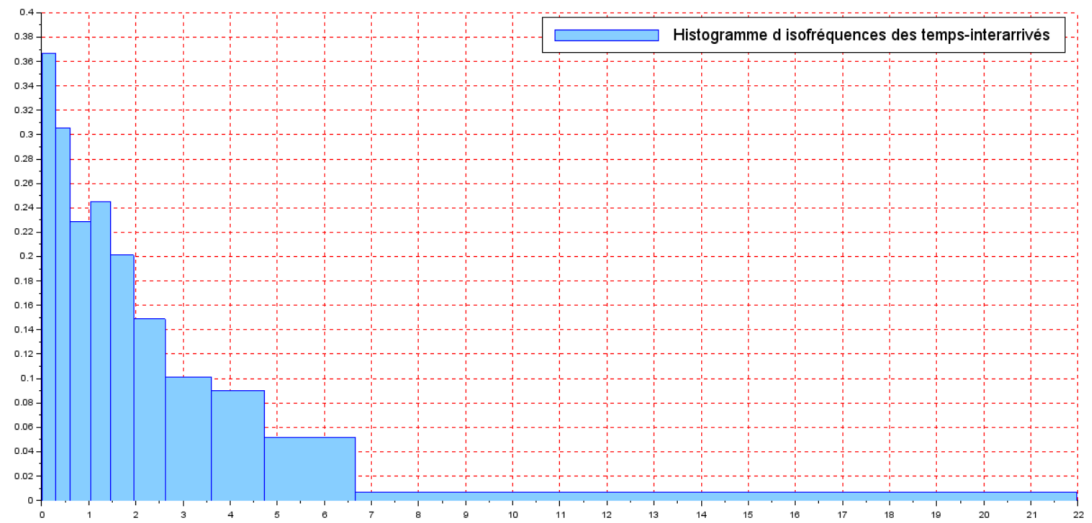
## 1.3 Histogramme

### 1.3.1 Histogramme avec classes isoamplitudes



Cet histogramme indique la concentration des données au sein des différentes classes de même amplitudes. On remarque à l'aide de cet histogramme que les valeurs sont principalement concentrées dans la première classe, qui regroupe les temps d'inter-arrivées de 0 à 2,15 secondes. Ensuite la concentration des valeurs est divisée d'environ par deux pour chacune des classes suivantes.

### 1.3.2 Histogramme avec classes isofréquences



Cet histogramme indique la concentration des données au sein des différentes classes de même fréquence. Nous visualisons rapidement la zone creuse lorsque les temps d'inter-arrivées dépassent les 6,30secondes.

## Deuxième partie

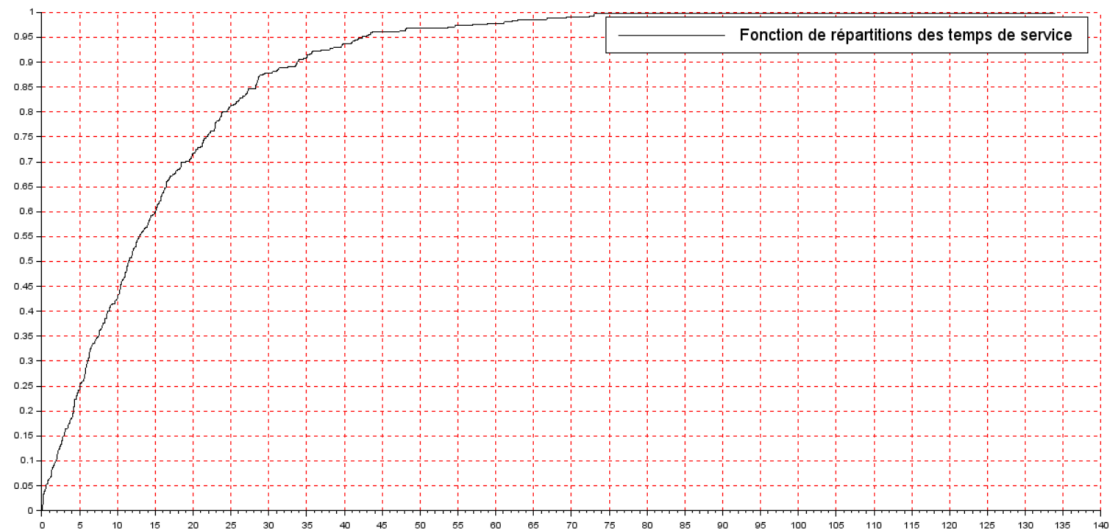
# Etude statistique des temps de service

## 2 Indicateurs de position et de dispersion

Indicateurs	Serveur 1	Serveur 2	Serveur 3
Minimum	0.01	0.04	0.01
Maximum	134	88.9	68.6
Etendue	134	88.8	68.6
Moyenne	15.5	10.6	6.27
Médiane	11.5	6.82	4.35
Q1	5.05	3.29	1.75
Q3	21.9	13.9	8.36
IQ	16.8	10.6	6.61
Ecart-Type	15	11.3	6.85
Variance	225	127	46.9

## 3 Fonctions de répartition

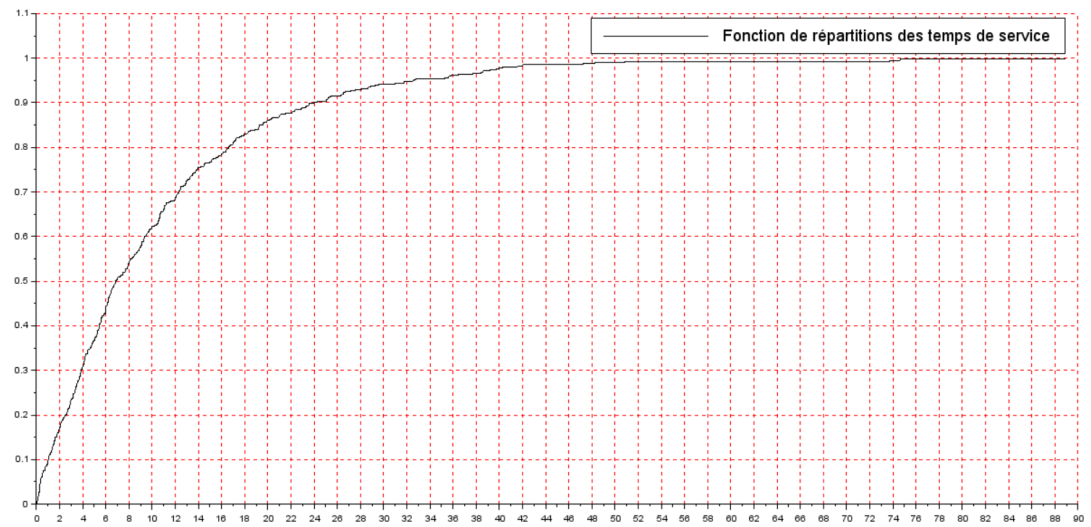
### 3.1 Serveur 1





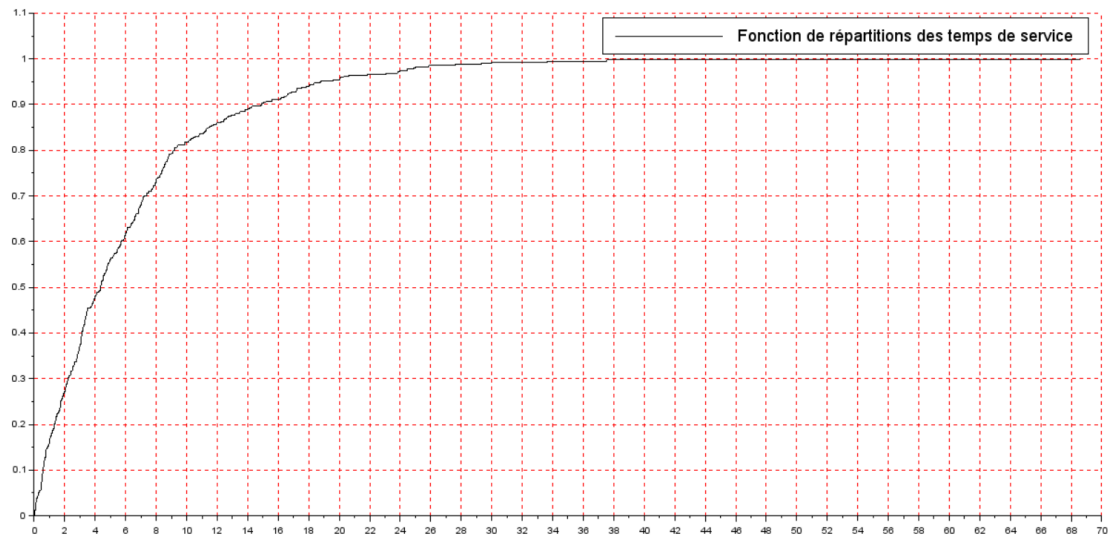
Ce graphique nous montre que la moitié des temps inter-arrivées au serveur 1 sont inférieurs ou égaux à 12 secondes.

### 3.2 Serveur 2



Ce graphique nous montre que la moitié des temps inter-arrivées au serveur 2 sont inférieurs ou égaux à 7 secondes.

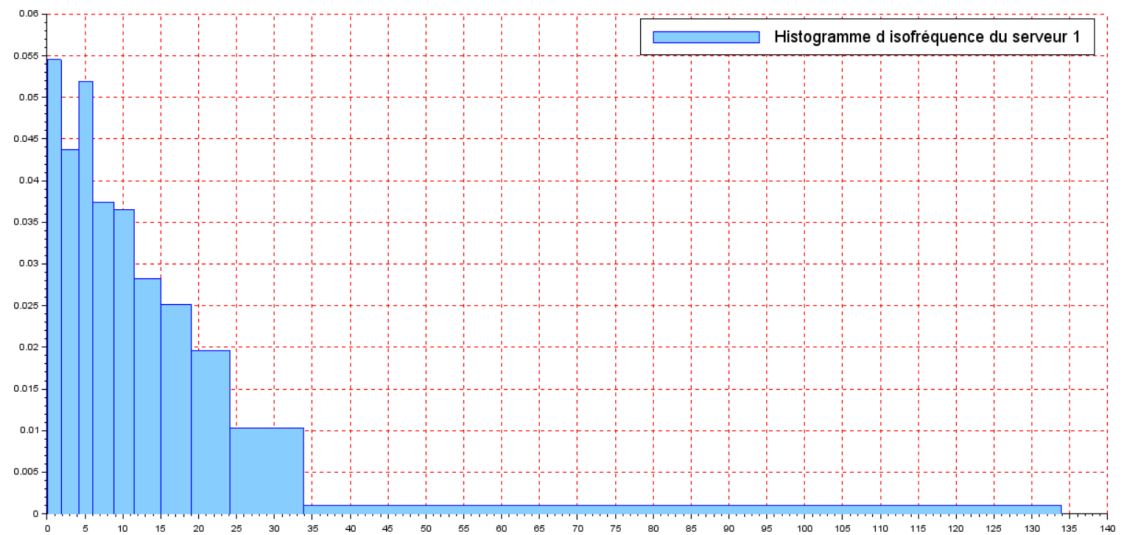
### 3.3 Serveur 3



Ce graphique nous montre que la moitié des temps inter-arrivées au serveur 3 sont inférieurs ou égaux à 4,3 secondes.

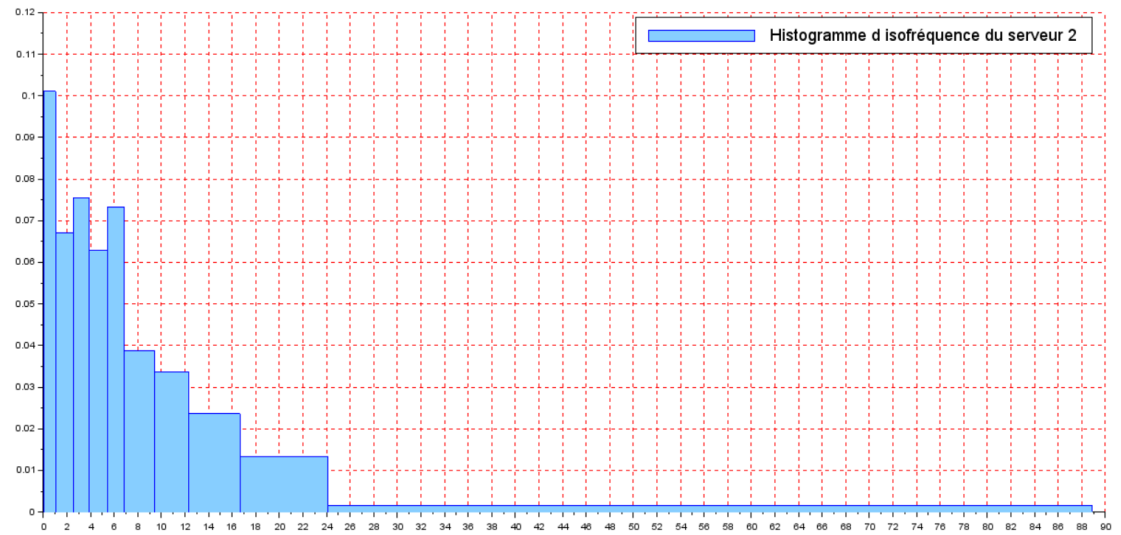
## 4 Histogrammes

### 4.1 Serveur 1



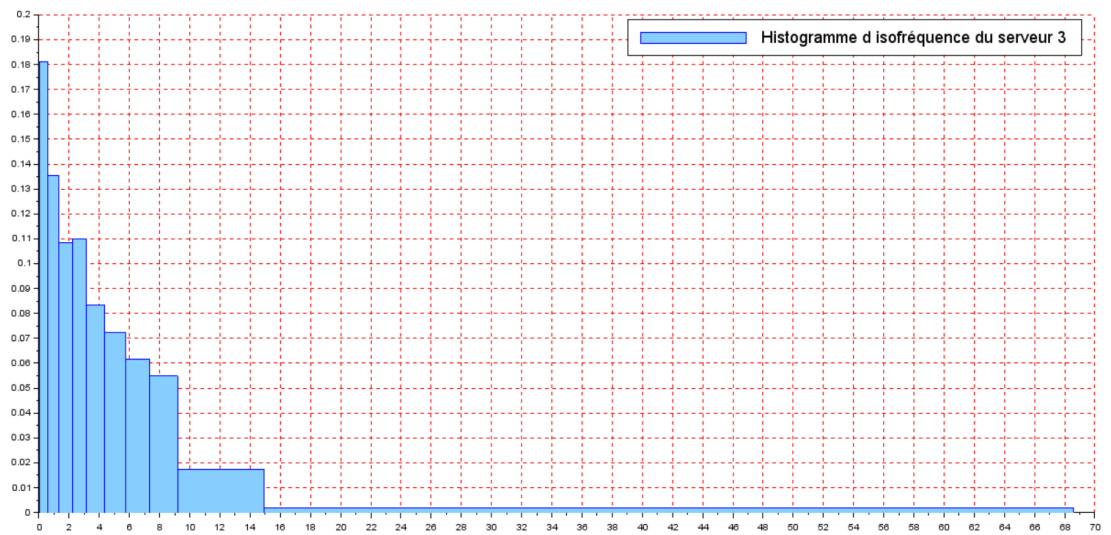
La zone creuse est présente pour les temps d'inter-arrivées supérieurs ou égaux à 34 secondes.

## 4.2 Serveur 2



La zone creuse est présente pour les temps d'inter-arrivées supérieurs ou égaux à 24 secondes.

## 4.3 Serveur 3



La zone creuse est présente pour les temps d'inter-arrivées supérieurs ou égaux à 15 secondes.

### Troisième partie

## Ajustement graphique à des lois mathématiques

### 5 Estimation des paramètres

#### 5.1 Loi normale

La loi normale  $X \sim N(\mu, \sigma)$  s'exprime en fonction de l'espérance  $\mu$  ainsi que de l'écart-type  $\sigma$ . On associe donc l'espérance à la moyenne des temps d'attentes/temps interarrivés.

#### 5.2 Loi uniforme

La loi uniforme dépend de 2 paramètres :  $a$  et  $b$  qui correspondent à l'intervalle  $[a, b]$  sur laquelle est définie la loi uniforme. On associe donc  $a$  au minimum et  $b$  au maximum des temps d'attentes/temps interarrivés.

#### 5.3 Loi exponentielle

La loi exponentielle dépend d'un seul paramètre :  $\lambda$ . Or on sait que pour une loi exponentielle  $X$  :

$$E(X) = \frac{1}{\lambda} \quad (1)$$

$$V(X) = \frac{1}{\lambda^2} \quad (2)$$

Donc

$$\lambda = \frac{1}{E(X)} \quad (3)$$

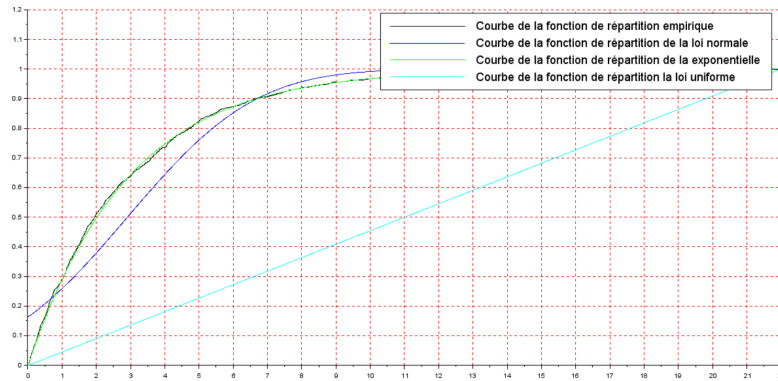
$$\lambda = \frac{1}{\sqrt{V(X)}} \quad (4)$$

$$= \frac{1}{\sigma} \quad (5)$$

On en déduit déjà que pour qu'une la moyenne doit être égale à l'écart type pour que les temps de services soient les réalisations d'une loi exponentielle.

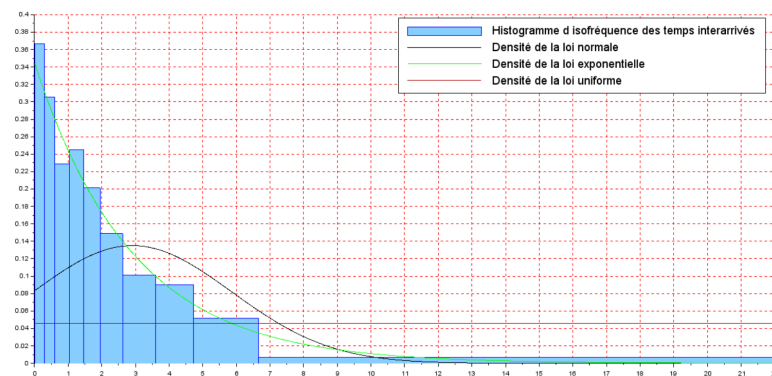
## 6 Tous les serveurs

### 6.1 Superposition des fonctions de répartition



Les courbes des fonctions de répartition de la loi uniforme et de la loi normale ne correspondent pas à celle des serveurs car l'une se comporte comme une droite et l'autre ne démarre pas avec sa vitesse d'accroissement maximale. Les courbes des fonctions de répartition empirique et de la loi exponentielle se superposent et ressemblent fortement à la fonction de répartition des serveurs.

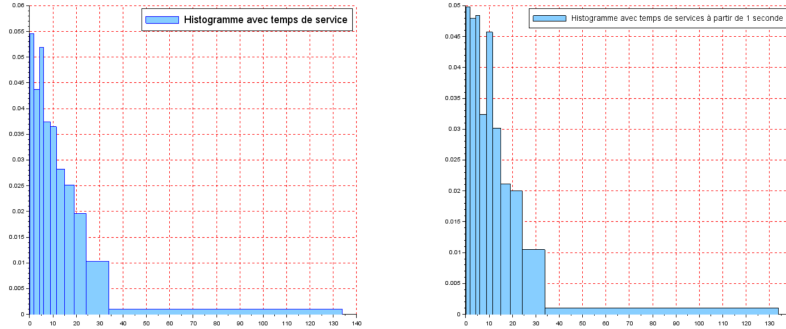
### 6.2 Superposition des fonctions de densité et de l'histogramme



Une loi uniforme a une étendue proche de 0 or on remarque que l'étendu des temps de services est largement supérieure à 0. On en conclut donc l'on ne peut modéliser les temps de services par une loi uniforme. On remarque aussi que la courbe de la loi normale n'est pas du tout proche de l'histogramme des temps de service, par contre la courbe de la fonction de densité de la loi exponentielle

s'approche de la forme de l'histogramme. De plus on remarque que la moyenne est presque égale à son écart-type, ce qui est une des caractéristiques de la loi exponentielle.

### 6.3 Propriété d'absence de mémoire

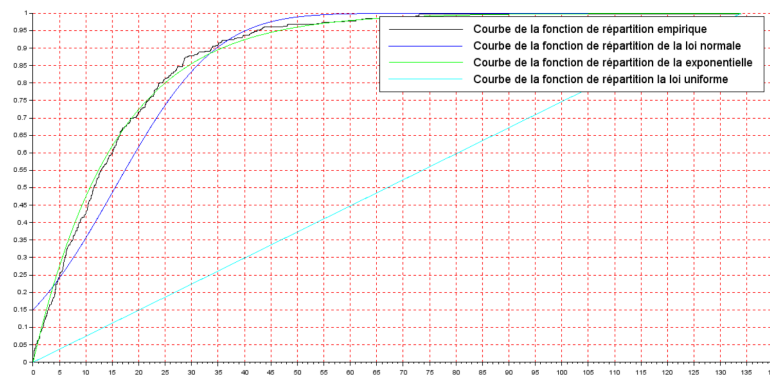


Indicateurs	Moyenne	Mediane	Variance	Ecart-type	Etendue
Temps interarrivés	2.9	1.95	8.73	2.95	22
Temps interarrivés après 1s	2.92	2.06	8.79	2.96	21
Différence	0.01	0.11	0.05	0.01	1

Nous remarquons aisément que ces deux histogrammes sont similaires, ce qui signifie que la différence entre les temps de services et les temps de services-1 seconde est proche de 0. Donc on en déduit qu'il y a une absence de mémoire.

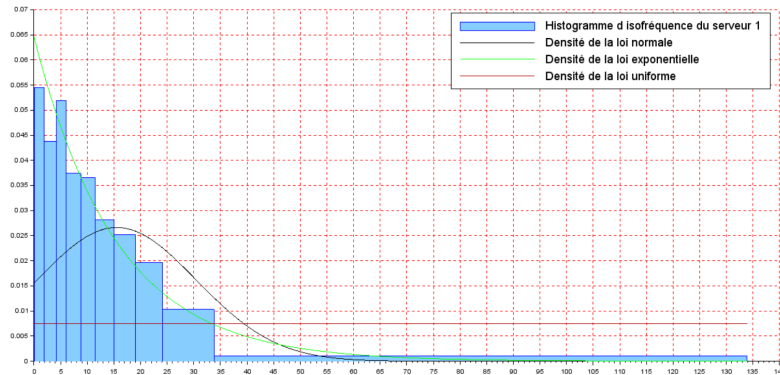
## 7 Serveur 1

### 7.1 Superposition des fonctions de répartition



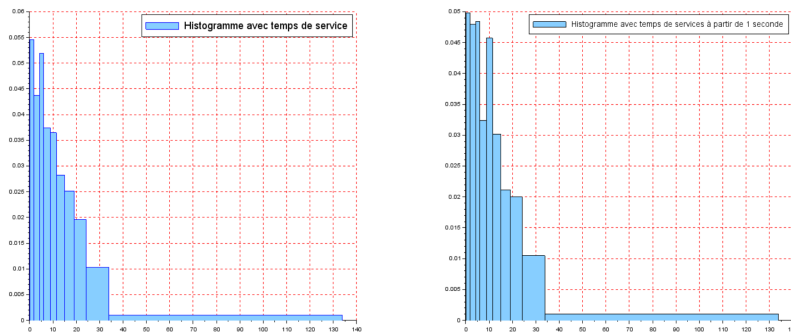
On remarque que la fonction de répartition de la loi exponentielle est celle qui correspond le plus pour représenter la répartition empirique des temps de service.

## 7.2 Superposition des fonctions de densité et de l'histogramme



Une loi uniforme a une étendue proche de 0 or on remarque que l'étendu des temps de services est largement supérieure à 0. On en conclut donc l'on ne peut modéliser les temps de services par une loi uniforme. On remarque aussi que la courbe de la loi normale n'est pas du tout proche de l'histogramme des temps de service, par contre la courbe de la fonction de densité de la loi exponentielle s'approche de la forme de l'histogramme. De plus on remarque que la moyenne est presque égale à son écart-type, ce qui est une des caractéristiques de la loi exponentielle.

## 7.3 Propriété d'absence de mémoire



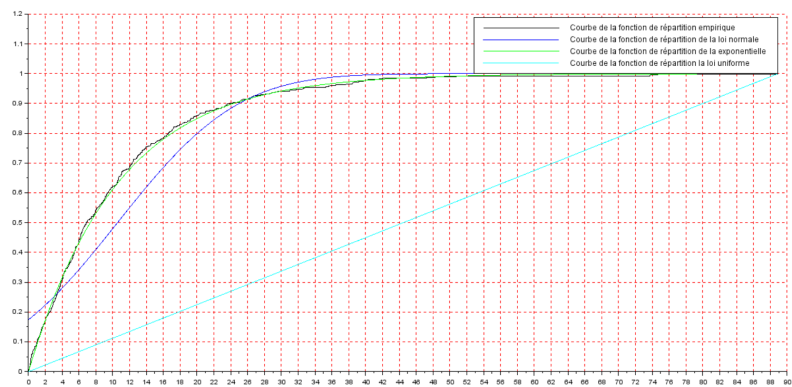


Indicateurs	Moyenne	Mediane	Variance	Ecart-type	Etendue
Temps d'attentes	15.50	11.49	224.59	14.98	133.89
Temps d'attentes après 1s	15.56	11.41	223.15	14.93	132.78
Différence	0.06	0.08	1.44	0.04	1.11

Nous remarquons aisément que ces deux histogrammes sont similaires, ce qui signifie que la différence entre les temps de services et les temps de services-1 seconde est proche de 0. Donc on en déduit qu'il y a une absence de mémoire.

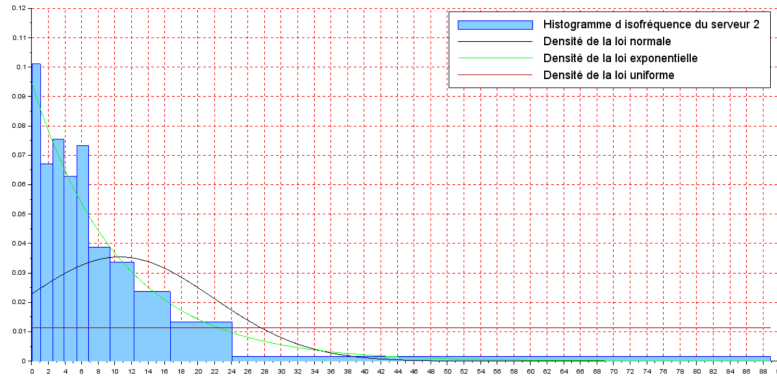
## 8 Serveur 2

### 8.1 Superposition des fonctions de répartition



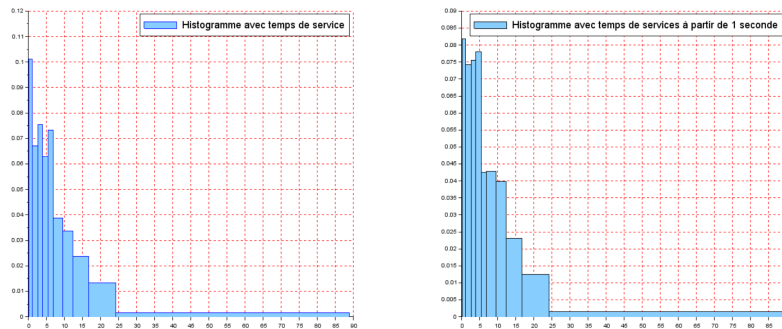
On remarque que la fonction de répartition de la loi exponentielle est celle qui correspond le plus pour représenter la répartition empirique des temps de service.

## 8.2 Superposition des fonctions de densité et de l'histogramme



Une loi uniforme a une étendue proche de 0 or on remarque que l'étendu des temps de services est largement supérieure à 0. On en conclut donc l'on ne peut modéliser les temps de services par une loi uniforme. On remarque aussi que la courbe de la loi normale n'est pas du tout proche de l'histogramme des temps de service, par contre la courbe de la fonction de densité de la loi exponentielle s'approche de la forme de l'histogramme. De plus on remarque que la moyenne est presque égale à son écart-type, ce qui est une des caractéristiques de la loi exponentielle.

## 8.3 Propriété d'absence de mémoire

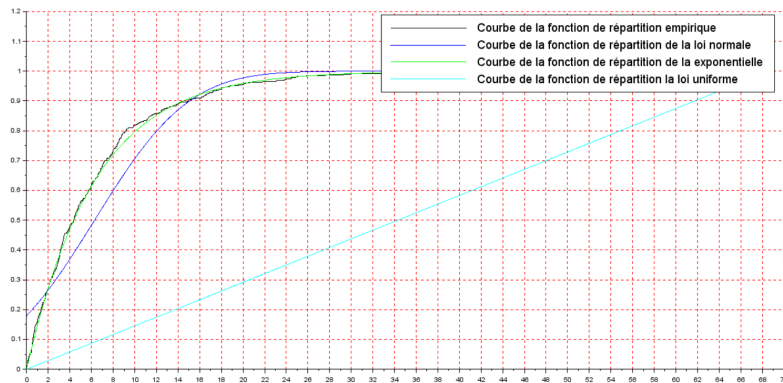


Indicateurs	Moyenne	Mediane	Variance	Ecart-type	Etendue
Temps d'attentes	10.56	6.82	126.84	11.26	88.85
Temps d'attentes après 1s	10.65	7.08	128.22	11.32	87.88
Différence	0.08	0.26	1.38	0.06	0.97

Nous remarquons aisément que ces deux histogrammes sont similaires, ce qui signifie que la différence entre les temps de services et les temps de services-1 seconde est proche de 0. Donc on en déduit qu'il y a une absence de mémoire.

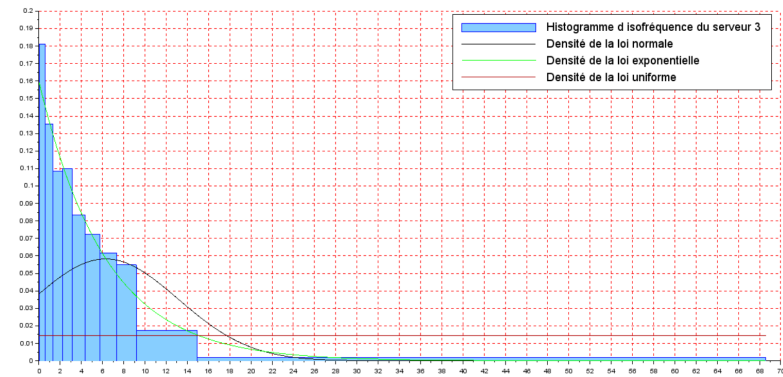
## 9 Serveur 3

### 9.1 Superposition des fonctions de répartition



On remarque que la fonction de répartition de la loi exponentielle est celle qui correspond le plus pour représenter la répartition empirique des temps de service.

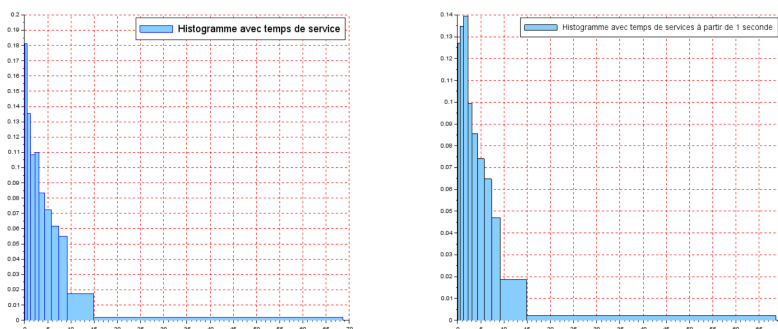
### 9.2 Superposition des fonctions de densité et de l'histogramme



Une loi uniforme a une étendue proche de 0 or on remarque que l'étendu des temps de services est largement supérieure à 0. On en conclut donc l'on ne peut modéliser les temps de services par une loi uniforme. On remarque aussi que la

courbe de la loi normale n'est pas du tout proche de l'histogramme des temps de service, par contre la courbe de la fonction de densité de la loi exponentielle s'approche de la forme de l'histogramme. De plus on remarque que la moyenne est presque égale à son écart-type, ce qui est une des caractéristiques de la loi exponentielle.

### 9.3 Propriété d'absence de mémoire



Indicateurs	Moyenne	Mediane	Variance	Ecart-type	Etendue
Temps d'attentes	6.27	4.35	46.88	6.84	68.58
Temps d'attentes après 1s	6.41	4.45	48.19	6.94	67.57
Différence	0.14	0.10	1.30	0.09	1.01

Nous remarquons aisément que ces deux histogrammes sont similaires, ce qui signifie que la différence entre les temps de services et les temps de services-1 seconde est proche de 0. Donc on en déduit qu'il y a une absence de mémoire.

## Quatrième partie

# Conclusion

A partir de l'étude des données fournies et grâce aux comparaisons avec des lois connues, nous avons pu déterminer que la loi exponentielle, de paramètre unique ( $\lambda = \frac{1}{\text{moyenne}}$ ) était celle qui se rapprochait le plus de ce qu'il se passe en réalité dans les serveurs, et donc qu'elle est la plus adaptée pour des simulations. En effet, lorsqu'on compare la fonction de répartition de la loi exponentielle à celles empiriques, et lorsqu'on compare la densité de la loi exponentielle avec les histogrammes des données, on remarque de très fortes ressemblances, beaucoup plus que pour la loi uniforme ou la loi normale.

De plus, on a remarqué empiriquement que le temps d'attente moyen ne changeait pas en fonction du temps qu'on avait déjà passé à attendre. Par exemple

pour un temps moyen de 5s, après 2s passées à attendre le temps moyen d'attente est toujours de 5s. Or, la loi exponentielle est une loi sans mémoire, c'est à dire que ses valeurs ne sont pas influencées par la valeurs précédentes, ce qui colle parfaitement avec le comportement de nos données.

Nous préconisons donc l'utilisation de cette loi exponentielle pour la reproduction de données réalistes et fiables dans l'optique d'une étude et d'une analyse poussées du système.

## A Etude statistique des temps interarrivés

### A.1 Indicateurs de position et de dispersion

```
// Extraction des temps inter-arrivées
t_ia = data(2:$, 2) - data(1:1237, 2);

extremes = [min(t_ia), max(t_ia)] // calcul du min et du max
moyenne = mean(t_ia) // calcul de la moyenne
mediane = perctl(t_ia,50) // calcul de la mediane

// calcul de la variance et de l'écart-type
v = variance(t_ia)
s = stdev(t_ia)

// calcul de l'étendue
etendue = extremes(2) - extremes(1)

Q1 = perctl(t_ia, 25) // premier quartile
Q3 = perctl(t_ia, 75) // troisième quartile
IQ = Q3(1) - Q1(1) // intervalle interquartile
```

### A.2 Fonction de répartition

```
// Extraction des temps inter-arrivées
t_ia = data(2:$, 2) - data(1:1237, 2);

tab = tabul(t_ia, 'i'); // construction du tableau des effectifs
tab(:,2) = tab(:,2)/length(t_ia); // calcul des fréquences
F = cumsum(tab(:,2)); // calcul des fréquences cumulées
plot2d2(tab(:,1),F)
legend("Fonction de répartitions des temps interarrivés")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];
```

### A.3 Histogramme

#### A.3.1 Histogramme avec classes isoamplitudes

```
// Extraction des temps inter-arrivées
t_ia = data(2:$, 2) - data(1:1237, 2);

C = linspace(min(t_ia), max(t_ia), 11) // calcul des classes
```

```
histplot(C, t_ia, style=2) // dessine l'histogramme
legend("Histogramme d isoamplitude des temps-interarrivés")
```

### A.3.2 Histogramme avec classes isofréquences

```
// Extraction des temps inter-arrivées
t_ia = data(2:$, 2) - data(1:1237, 2);

deciles=perctl(t_ia,10:10:90) // Calcul des déciles
// Affectations d'isofréquences comme bornes de classes
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_ia)
ClassesDeciles(11)=max(t_ia)

histplot(ClassesDeciles,t_ia,style=2) // dessine l'histogramme
legend("Histogramme d isofréquences des temps-interarrivés")
```

## B Etude statistique des temps de service

### B.1 Indicateurs de position et de dispersion

#### B.1.1 Serveur 1

```
// Extraction des temps de service
index_bool = ( data(:, 3) = 1 )
tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4)

extremesS1 = [min(t_s1), max(t_s1)] // calcul du min et du max
moyenneS1 = mean(t_s1) // calcul de la moyenne
medianeS1 = perctl(t_s1,50) // calcul de la mediane

// calcul de la variance et de l'écart-type
vS1 = variance(t_s1)
sS1 = stdev(t_s1)

// calcul de l'étendue
etendueS1 = extremesS1(2) - extremesS1(1)

Q1S1 = perctl(t_s1, 25) // premier quartile
Q3S1 = perctl(t_s1, 75) // troisième quartile
IQS1 = Q3S1(1) - Q1S1(1) // intervalle interquartile
```

### B.1.2 Serveur 2

```
// Extraction des temps de service
index_bool = ( data(:, 3) = 2 )
tabs2 = data(index_bool, :)
t_s2 = tabs2(1:$,4)

extremesS2 = [min(t_s2), max(t_s2)] // calcul du min et du max
moyenneS2 = mean(t_s2) // calcul de la moyenne
medianeS2 = perctl(t_s2,50) // calcul de la mediane

// calcul de la variance et de l'écart-type
vS2 = variance(t_s2)
sS2 = stdev(t_s2)

// calcul de l'étendue
etendueS2 = extremesS2(2) - extremesS2(1)

Q1S2 = perctl(t_s2, 25) // premier quartile
Q3S2 = perctl(t_s2, 75) // troisième quartile
IQS2 = Q3S2(1) - Q1S2(1) // intervalle interquartile
```

### B.1.3 Serveur 3

```
// Extraction des temps de service
index_bool = ( data(:, 3) = 3 )
tabS3 = data(index_bool, :)
t_s3 = tabS3(1:$,4)

extremesS3 = [min(t_s3), max(t_s3)] // calcul du min et du max
moyenneS3 = mean(t_s3) // calcul de la moyenne
medianeS3 = perctl(t_s3,50) // calcul de la mediane

// calcul de la variance et de l'écart-type
vS3 = variance(t_s3)
sS3 = stdev(t_s3)

// calcul de l'étendue
etendueS3 = extremesS3(2) - extremesS3(1)

Q1S3 = perctl(t_s3, 25) // premier quartile
Q3S3 = perctl(t_s3, 75) // troisième quartile
IQS3 = Q3S3(1) - Q1S3(1) // intervalle interquartile
```



## B.2 Fonctions de repartitions

### B.2.1 Serveur 1

```
// Extraction des temps de service

index_bool = ( data(:, 3) == 1 )
tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4);

tab = tabul(t_s1,'i')
tab(:,2) = tab(:,2)/length(t_s1)
F = cumsum(tab(:,2))

plot2d2(tab(:,1),F)
legend("Fonction de répartitions des temps de service")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];
```

### B.2.2 Serveur 2

```
// Extraction des temps de service

index_bool = ( data(:, 3) == 2 )
tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4);

tab = tabul(t_s1,'i')
tab(:,2) = tab(:,2)/length(t_s1)
F = cumsum(tab(:,2))

plot2d2(tab(:,1),F)
legend("Fonction de répartitions des temps de service")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];
```

### B.2.3 Serveur 3

```
// Extraction des temps de service

index_bool = ( data(:, 3) == 3 )
```

```

tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4);

tab = tabul(t_s1,'i')
tab(:,2) = tab(:,2)/length(t_s1)
F = cumsum(tab(:,2))

plot2d2(tab(:,1),F)
legend("Fonction de répartitions des temps de service")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

## B.3 Histogrammes

### B.3.1 Serveur 1

```

// Extraction des temps de service
index_bool = ( data(:, 3) == 1 )
tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4);

deciles=perctl(t_s1,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s1)
ClassesDeciles(11)=max(t_s1)

histplot(ClassesDeciles,t_s1,style=2)
legend("Histogramme d isofréquence du serveur 1")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

### B.3.2 Serveur 2

```

// Extraction des temps de service
index_bool = ( data(:, 3) == 2 )
tabS2 = data(index_bool, :)
t_s2 = tabS2(1:$,4);

```

```

deciles=perctl(t_s2,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s2)
ClassesDeciles(11)=max(t_s2)

histplot(ClassesDeciles,t_s2,style=2)
legend("Histogramme d isofréquence du serveur 2")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

### B.3.3 Serveur 3

```

// Extraction des temps de service
index_bool = ( data(:, 3) == 3 )
tabS3 = data(index_bool, :)
t_s3 = tabS3(1:$,4);

deciles=perctl(t_s3,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s3)
ClassesDeciles(11)=max(t_s3)

histplot(ClassesDeciles,t_s3,style=2)
legend("Histogramme d isofréquence du serveur 3")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

## C Ajustement graphique à des lois mathématiques

### C.1 Tous les serveurs

#### C.1.1 Superposition des fonctions de répartitions

```

// Extraction des temps inter-arrivées
t_ia = data(2:$, 2) - data(1:1237, 2);

```

```

tab = tabul(t_ia, 'i'); // construction du tableau des effectifs
tab(:,2) = tab(:,2)/length(t_ia); // calcul des fréquences
F = cumsum(tab(:,2)); // calcul des fréquences cumulées
plot2d2(tab(:,1),F)
legend("Fonction de répartitions des temps interarrivés")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

// Répartition loi normale
a=min(t_ia):0.01:max(t_ia)
m=ones(a)*mean(t_ia)
s=ones(a)*stdev(t_ia)
[P,Q]=cdfnor("PQ",a,m,s)
plot2d2(a,P,style=2)

// Repartition loi exponentielle
lambda=1/mean(t_ia)
b=1-exp(-lambda*a)
plot2d2(a,b,style=3)

// Repartition loi uniforme
c=(a-min(t_ia))/(max(t_ia)-min(t_ia))
plot2d2(a,c,style=4)

legend("Courbe de la fonction de répartition empirique","Courbe de la fonction
de répartition de la loi normale","Courbe de la fonction de répartition de la
exponentielle","Courbe de la fonction de répartition la loi uniforme")

```

### C.1.2 Superposition des fonctions de densité et de l'histogramme

```

// Extraction des temps inter-arrivées
t_ia = data(2:$, 2) - data(1:1237, 2);

// Histogramme
deciles=perctl(t_ia,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_ia)
ClassesDeciles(11)=max(t_ia)
histplot(ClassesDeciles,t_ia,style=2)

// Densité de la loi normale

```

```

a=min(t_ia):0.01:max(t_ia)
m=mean(t_ia)
v=stdev(t_ia)
b=(1/(v*sqrt(2*%pi))*exp((-1/2)*((a-m)/v)^2))
plot2d2(a,b,style=1)

// Densité de la loi exponentielle
lambda=1/mean(t_ia)
b=lambda*exp(-lambda*a)
plot2d2(a,b,style=3)

// Densité de la loi uniforme
h=1/(max(t_ia)-min(t_ia))
b=ones(a)*h
plot2d2(a,b,style=20)

legend("Histogramme d isofréquence des temps interarrivés","Densité de la loi
normale","Densité de la loi exponentielle","Densité de la loi uniforme")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

### C.1.3 Propriété d'absence de mémoire

```

extremesS = [min(t_ia), max(t_ia)] // calcul du min et du max
moyenneS = mean(t_ia) // calcul de la moyenne
medianeS = perctl(t_ia,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
vS = variance(t_ia)
sS = stdev(t_ia)
// calcul de l'étendue
etendueS = extremesS(2) - extremesS(1)
Q1S = perctl(t_ia, 25) // premier quartile
Q3S = perctl(t_ia, 75) // troisième quartile
IQS = Q3S(1) - Q1S(1) // intervalle interquartile

index_bool3 = (t_ia > 1)
tab_ia = t_ia(index_bool3)
new_t_ia = tab_ia - 1
extremes_ia = [min(new_t_ia), max(new_t_ia)] // calcul du min et du max
moyenne_ia = mean(new_t_ia) // calcul de la moyenne
mediane_ia = perctl(new_t_ia,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
v = variance(new_t_ia)

```

```

e = stdev(new_t_ia)
// calcul de l'étendue
etendue = extremes_ia(2) - extremes_ia(1)
Q1_ = perctl(new_t_ia, 25) // premier quartile
Q3_ = perctl(new_t_ia, 75) // troisième quartile
IQ = Q3_(1) - Q1_(1) // intervalle interquartile

comparaison = [moyenne_ia - moyenneS, mediane_ia(1) - medianeS(1), v - vS, e - sS, etendue - sS]

subplot(1,2,1)
deciles=perctl(t_ia,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_ia)
ClassesDeciles(11)=max(t_ia)
histplot(ClassesDeciles,t_ia,style=2)
legend("Histogramme avec temps de service")

subplot(1,2,2)
histplot(ClassesDeciles,new_t_ia,style=1)
legend("Histogramme avec temps de services à partir de 1 seconde")

```

## C.2 Serveur 1

### C.2.1 Superposition des fonctions de répartitions

```

// Extraction des temps de service

index_bool = ( data(:, 3) == 1 )
tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4);

// Répartition empirique
tab = tabul(t_s1,'i')
tab(:,2) = tab(:,2)/length(t_s1)
F = cumsum(tab(:,2))

plot2d2(tab(:,1),F)

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

// Répartition loi normale

```

```

a=min(t_s1):0.01:max(t_s1)
m=ones(a)*mean(t_s1)
s=ones(a)*stdev(t_s1)
[P,Q]=cdfnor("PQ",a,m,s)
plot2d2(a,P,style=2)

// Repartition loi exponentielle
lambda=1/mean(t_s1)
b=1-exp(-lambda*a)
plot2d2(a,b,style=3)

// Repartition loi uniforme
c=(a-min(t_s1))/(max(t_s1)-min(t_s1))
plot2d2(a,c,style=4)

legend("Courbe de la fonction de répartition empirique","Courbe de la fonction
de répartition de la loi normale","Courbe de la fonction de répartition de la
exponentielle","Courbe de la fonction de répartition la loi uniforme")

```

### C.2.2 Superposition des fonctions de densité et de l'histogramme

```

// Extraction des temps de service
index_bool = ( data(:, 3) == 1 )
tabS1 = data(index_bool, :)
t_s1 = tabS1(1:$,4);

deciles=perctl(t_s1,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s1)
ClassesDeciles(11)=max(t_s1)

histplot(ClassesDeciles,t_s1,style=2)

// Densité de la loi normale
a=min(t_s1):0.01:max(t_s1)
m=mean(t_s1)
v=stdev(t_s1)
b=(1/(v*sqrt(2*\%pi)))*exp((-1/2)*((a-m)/v)^2)
plot2d2(a,b,style=1)

// Densité de la loi exponentielle
lambda=1/mean(t_s1)
b=lambda*exp(-lambda*a)
plot2d2(a,b,style=3)

```

```

// Densité de la loi uniforme
h=1/(max(t_s1)-min(t_s1))
b=ones(a)*h
plot2d2(a,b,style=20)

legend("Histogramme d isofréquence du serveur 1","Densité de la loi
normale","Densité de la loi exponentielle","Densité de la loi uniforme")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

### C.2.3 Propriété d'absence de mémoire

```

extremesS1 = [min(t_s1), max(t_s1)] // calcul du min et du max
moyenneS1 = mean(t_s1) // calcul de la moyenne
medianeS1 = perctl(t_s1,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
vS1 = variance(t_s1)
sS1 = stdev(t_s1)
// calcul de l'étendue
etendueS1 = extremesS1(2) - extremesS1(1)
Q1S1 = perctl(t_s1, 25) // premier quartile
Q3S1 = perctl(t_s1, 75) // troisième quartile
IQS1 = Q3S1(1) - Q1S1(1) // intervalle interquartile

index_bool1 = (t_s1 > 1);
tab_s1 = t_s1(index_bool1);
new_t_s1 = tab_s1 - 1;
extremes1 = [min(new_t_s1), max(new_t_s1)] // calcul du min et du max
moyenne1 = mean(new_t_s1) // calcul de la moyenne
mediane1 = perctl(new_t_s1,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
v1 = variance(new_t_s1)
e1 = stdev(new_t_s1)
// calcul de l'étendue
etendue1 = extremes1(2) - extremes1(1)
Q1_1 = perctl(new_t_s1, 25) // premier quartile
Q3_1 = perctl(new_t_s1, 75) // troisième quartile
IQ1 = Q3_1(1) - Q1_1(1) // intervalle interquartile

comparaison1 = [moyenne1 - moyenneS1, mediane1(1) - medianeS1(1), v1 - vS1, e1 - sS1, etendu

subplot(1,2,1)

```



```

deciles=perctl(t_s1,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s1)
ClassesDeciles(11)=max(t_s1)
histplot(ClassesDeciles,t_s1,style=2)
legend("Histogramme avec temps de service")

subplot(1,2,2)
histplot(ClassesDeciles,new_t_s1,style=1)
legend("Histogramme avec temps de services à partir de 1 seconde")

```

### C.3 Serveur 2

#### C.3.1 Superposition des fonctions de répartitions

```

// Extraction des temps de service

index_bool = ( data(:, 3) == 2 )
tabS2 = data(index_bool, :)
t_s2 = tabS2(1:$,4);

// Répartition empirique
tab = tabul(t_s2,'i')
tab(:,2) = tab(:,2)/length(t_s2)
F = cumsum(tab(:,2))

plot2d2(tab(:,1),F)

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

// Répartition loi normale
a=min(t_s2):0.01:max(t_s2)
m=ones(a)*mean(t_s2)
s=ones(a)*stdev(t_s2)
[P,Q]=cdfnor("PQ",a,m,s)
plot2d2(a,P,style=2)

// Répartition loi exponentielle
lambda=1/mean(t_s2)
b=1-exp(-lambda*a)
plot2d2(a,b,style=3)

```

```
// Repartition loi uniforme
c=(a-min(t_s2))/(max(t_s2)-min(t_s2))
plot2d2(a,c,style=4)

legend("Courbe de la fonction de répartition empirique","Courbe de la fonction
de répartition de la loi normale","Courbe de la fonction de répartition de la
exponentielle","Courbe de la fonction de répartition la loi uniforme")
```

### C.3.2 Superposition des fonctions de densité et de l'histogramme

```
// Extraction des temps de service
index_bool = ( data(:, 3) == 2 )
tabS2 = data(index_bool, :)
t_s2 = tabS2(1:$,4);

deciles=perctl(t_s2,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s2)
ClassesDeciles(11)=max(t_s2)

histplot(ClassesDeciles,t_s2,style=2)

// Densité de la loi normale
a=min(t_s2):0.01:max(t_s2)
m=mean(t_s2)
v=stdev(t_s2)
b=(1/(v*sqrt(2*\%pi)))*exp((-1/2)*((a-m)/v)^2))
plot2d2(a,b,style=1)

// Densité de la loi exponentielle
lambda=1/mean(t_s2)
b=lambda*exp(-lambda*a)
plot2d2(a,b,style=3)

// Densité de la loi uniforme
h=1/(max(t_s2)-min(t_s2))
b=ones(a)*h
plot2d2(a,b,style=20)

legend("Histogramme d isofréquence du serveur 2","Densité de la loi
normale","Densité de la loi exponentielle","Densité de la loi uniforme")

// Définition des paramètres d'affichages
```

```

a=gca();
a.x_location = "origin";
a.grid=[5,5];

```

### C.3.3 Propriété d'absense de mémoire

```

extremesS2 = [min(t_s2), max(t_s2)] // calcul du min et du max
moyenneS2 = mean(t_s2) // calcul de la moyenne
medianeS2 = perctl(t_s2,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
vS2 = variance(t_s2)
sS2 = stdev(t_s2)
// calcul de l'étendue
etendueS2 = extremesS2(2) - extremesS2(1)
Q1S2 = perctl(t_s2, 25) // premier quartile
Q3S2 = perctl(t_s2, 75) // troisième quartile
IQS2 = Q3S2(1) - Q1S2(1) // intervalle interquartile

index_bool2 = (t_s2 > 1)
tab_s2 = t_s2(index_bool2)
new_t_s2 = tab_s2 - 1
extremes2 = [min(new_t_s2), max(new_t_s2)] // calcul du min et du max
moyenne2 = mean(new_t_s2) // calcul de la moyenne
mediane2 = perctl(new_t_s2,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
v2 = variance(new_t_s2)
e2 = stdev(new_t_s2)
// calcul de l'étendue
etendue2 = extremes2(2) - extremes2(1)
Q1_2 = perctl(new_t_s2, 25) // premier quartile
Q3_2 = perctl(new_t_s2, 75) // troisième quartile
IQ2 = Q3_2(1) - Q1_2(1) // intervalle interquartile

comparaison2 = [moyenne2 - moyenneS2, mediane2(1) - medianeS2(1), v2 - vS2, e2 - sS2, etendu

subplot(1,2,1)
deciles=perctl(t_s2,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s2)
ClassesDeciles(11)=max(t_s2)
histplot(ClassesDeciles,t_s2,style=2)
legend("Histogramme avec temps de service")

subplot(1,2,2)

```

```
histplot(ClassesDeciles,new_t_s2,style=1)
legend("Histogramme avec temps de services à partir de 1 seconde")
```

## C.4 Serveur 3

### C.4.1 Superposition des fonctions de répartitions

```
// Extraction des temps de service

index_bool = ( data(:, 3) == 3 )
tabS3 = data(index_bool, :)
t_s3 = tabS3(1:$,4);

// Repartition empirique
tab = tabul(t_s3,'i')
tab(:,2) = tab(:,2)/length(t_s3)
F = cumsum(tab(:,2))

plot2d2(tab(:,1),F)

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];

// Répartition loi normale
a=min(t_s3):0.01:max(t_s3)
m=ones(a)*mean(t_s3)
s=ones(a)*stdev(t_s3)
[P,Q]=cdfnor("PQ",a,m,s)
plot2d2(a,P,style=2)

// Repartition loi exponentielle
lambda=1/mean(t_s3)
b=1-exp(-lambda*a)
plot2d2(a,b,style=3)

// Repartition loi uniforme
c=(a-min(t_s3))/(max(t_s3)-min(t_s3))
plot2d2(a,c,style=4)

legend("Courbe de la fonction de répartition empirique","Courbe de la fonction
de répartition de la loi normale","Courbe de la fonction de répartition de la
exponentielle","Courbe de la fonction de répartition la loi uniforme")
```

#### C.4.2 Superposition des fonctions de densité et de l'histogramme

```
// Extraction des temps de service
index_bool = ( data(:, 3) == 3 )
tabS3 = data(index_bool, :)
t_s3 = tabS3(1:$,4);

deciles=perctl(t_s3,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s3)
ClassesDeciles(11)=max(t_s3)

histplot(ClassesDeciles,t_s3,style=2)

// Densité de la loi normale
a=min(t_s3):0.01:max(t_s3)
m=mean(t_s3)
v=stdev(t_s3)
b=(1/(v*sqrt(2*pi)))*exp((-1/2)*((a-m)/v)^2))
plot2d2(a,b,style=1)

// Densité de la loi exponentielle
lambda=1/mean(t_s3)
b=lambda*exp(-lambda*a)
plot2d2(a,b,style=3)

// Densité de la loi uniforme
h=1/(max(t_s3)-min(t_s3))
b=ones(a)*h
plot2d2(a,b,style=20)

legend("Histogramme d isofréquence du serveur 3","Densité de la loi
normale","Densité de la loi exponentielle","Densité de la loi uniforme")

// Définition des paramètres d'affichages
a=gca();
a.x_location = "origin";
a.grid=[5,5];
```

#### C.4.3 Propriété d'absence de mémoire

```
extremesS3 = [min(t_s3), max(t_s3)] // calcul du min et du max
moyenneS3 = mean(t_s3) // calcul de la moyenne
medianeS3 = perctl(t_s3,50) // calcul de la mediane
```

```

// calcul de la variance et de l'écart-type
vS3 = variance(t_s3)
sS3 = stdev(t_s3)
// calcul de l'étendue
etendueS3 = extremesS3(2) - extremesS3(1)
Q1S3 = perctl(t_s3, 25) // premier quartile
Q3S3 = perctl(t_s3, 75) // troisième quartile
IQS3 = Q3S3(1) - Q1S3(1) // intervalle interquartile

index_bool3 = (t_s3 > 1)
tab_s3 = t_s3(index_bool3)
new_t_s3 = tab_s3 - 1
extremes3 = [min(new_t_s3), max(new_t_s3)] // calcul du min et du max
moyenne3 = mean(new_t_s3) // calcul de la moyenne
mediane3 = perctl(new_t_s3,50) // calcul de la mediane
// calcul de la variance et de l'écart-type
v3 = variance(new_t_s3)
e3 = stdev(new_t_s3)
// calcul de l'étendue
etendue3 = extremes3(2) - extremes3(1)
Q1_3 = perctl(new_t_s3, 25) // premier quartile
Q3_3 = perctl(new_t_s3, 75) // troisième quartile
IQ3 = Q3_3(1) - Q1_3(1) // intervalle interquartile

comparaison3 = [moyenne3 - moyenneS3, mediane3(1) - medianeS3(1), v3 - vS3, e3 - sS3, etendu

subplot(1,2,1)
deciles=perctl(t_s3,10:10:90);
for i=2:10
    ClassesDeciles(i)=deciles(i-1)
end
ClassesDeciles(1)=min(t_s3)
ClassesDeciles(11)=max(t_s3)
histplot(ClassesDeciles,t_s3,style=2)
legend("Histogramme avec temps de service")

subplot(1,2,2)
histplot(ClassesDeciles,new_t_s3,style=1)
legend("Histogramme avec temps de services à partir de 1 seconde")

```