Projet de Fiabilité

Ugo Devoille & Ariane Ndong Mba

Novembre 2020

Contents

1 Introduction

Un système est constitué de plusieurs composants assurant divers fonctions. Une des plus importantes mesures de sa performance est sa fiabilité. La fiabilité d'un système est définie comme étant la probabilité que le système fonctionne durant une période de temps sous des conditions spécifiées. Un objectif de la théorie de la fiabilité est de trouver le moyen d'évaluer la fiabilité d'un système complexe à partir de la connaissance des fiabilités des composants le constituant. D'où l'évaluation de la fiabilité d'un système est une caractéristique importante.

Plus un système est constitué d'un nombre important de composants plus la fiabilité de ce dernier à tendance à diminuer. Lorsque les composants sont trop nombreux ou trop complexes, il arrive fréquemment un moment où la maîtrise de la fiabilité n'est plus possible et l'hypothèse d'une défaillance très probable. La recherche de la diminution du coût des défaillances en exploitation a donc entraîné une augmentation des exigences de fiabilité sur les systèmes.

Dans ce projet, nous allons mettre en pratique les notions de fiabilité vues en cours sur des systèmes complexes non-réparables et réparables.En premier, nous allons présenter les résultats des travaux pratiques faits en cours et en second faire une étude guidée d'un système complexe précis choisis par nous.

2 Compte-Rendu du TP1

2.1 Fonction de structure

Le système complexe étudié ici est le suivant :

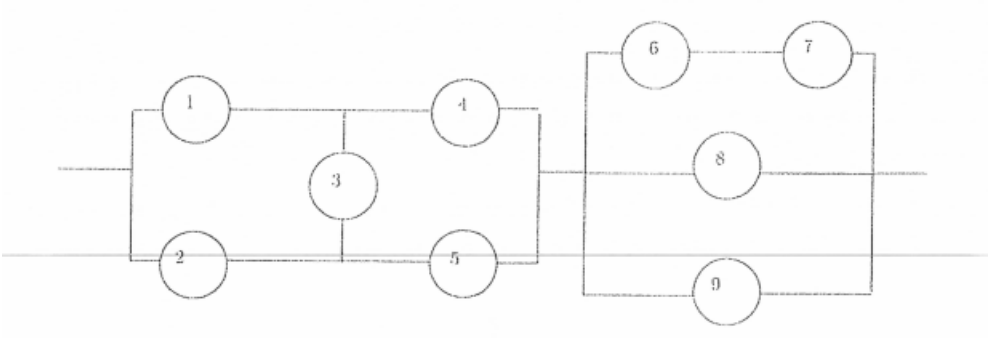


Figure 1: Système représenté

On remarque que ce système est décomposable en deux gros blocs, celui de gauche et celui de droite. Le bloc de gauche est constitué des états numérotés de 1 à 5, tandis que celui de droite comporte les états numérotés de 6 à 9. En nommant $\Phi_1(x)$ et $\Phi_2(x)$ les fonctions caractéristiques du bloc de droite et du bloc de gauche respectivement, et $\Phi(x)$ la fonction caractéristique globale du système, on à alors l'équation :

$$\Phi_1(x) * \Phi_2(x) = \Phi(x)$$

car ces deux blocs sont placés en série.

Trouvons donc maintenant une formulation pour $\Phi_1(x)$. On remarque que deux cas existent dans le bloc de gauche; celui où la composante 3 fonctionne, et celui où elle ne fonctionne pas. Ainsi, on peut écrire sa fonction caractéristique de la manière suivante :

$$\Phi_1(x) = x_3[1-(1-x_1)(1-x_2)][1-(1-x_4)(1-x_5)] + (1-x_3)[(1-(1-x_1x_4))(1-(1-x_2x_5))]$$

Pour le bloc de droite, on a un système Série/ Parallèle assez simple à définir. Ainsi, sa fonction caractéristique s'écrit sous la forme :

$$\Phi_2(x) = 1-(1-x_6x_7)(1-x_8)(1-x_9)$$

Grâce à ces informations, il est possible d'implémenter dans R la fonction caractéristique du système dans son ensemble. On met une variable en plus sys qui est de 0 par définition, et dans ce cas on renvoit $\Phi(x)$. Cependant, si sys=1, on renvoit $\Phi_1(x)$, et si sys=2, on renvoit $\Phi_2(x)$. Il est possible de trouver le code de la fonction dans l'annexe.

Nous faisons ensuite quelques exemples montrons que le système marche correctement. Pour plus de clarté, on associe un schéma où les états entourés sont ceux qui marchent, et on indique le chemin si il en existe un. En entrant le vecteur $x = (1,1,1,0,0,1,1,0,1)$, on obtient :

```
## [1] 0
```

On a donc un premier système qui ne marche pas. Le second système est lui défini pour être en marche.

Le vecteur utilisé ici est x = (1,0,1,0,1,1,1,0,0). La fonction renvoie :

```
## [1] 1
```

Comme le montre le schéma ainsi que la sortie R, pour ce vecteur, le système marche.

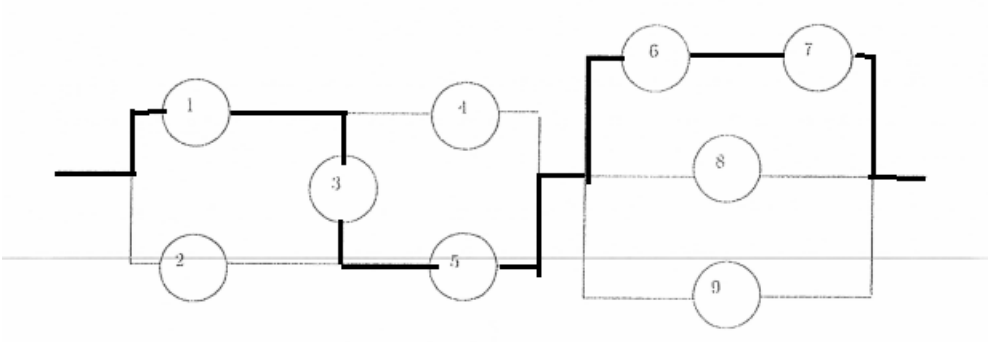


Figure 2: Système qui marche

2.2 Fonction de survie

La fonction de survie est définie en fiabilité comme étant :

$$S(t) = P(T > t)$$

avec t le temps, et T est la V.A. qui symbolise l'instant de décès (ou plutôt de panne ici). Pour un système que l'on note $R_S(t)$, on peut associer sa fonction de survie en la définissant par :

$$R_S(t) = \Phi(R_c)$$

avec R_c le vecteur composé des fonctions de survie pour chaque composante du système. On notera $\bar{F}(t)$ la fonction de survie aléatoire qui modélise la durée de vie du système dans le cadre de ce TP. On notera que cela s'implémente facilement dans R puisqu'il suffit de changer la variable d'entrée dans la fonction phi, ce qui est facilement adaptable.

2.2.1 Fonction de survie avec composantes suivant une loi de Weibull.

La fonction de Weibull connait une grande utilisation dans le domaine de la fiabilité, pour l'analyse de la durée de vie, puisque celle ci est très flexible. En notant $\lambda > 0$ le paramètre d'échelle et $\beta > 0$ le paramètre de forme, on peut définit la densité de la loi de Weibull par :

$$f(t, \lambda, \beta) = \frac{\beta}{\lambda}\left(\frac{t}{\lambda}\right)^{\beta-1} e^{-\left(\frac{t}{\lambda}\right)^{\beta}}$$

Ensuite, il est aussi possible d'obtenir la fonction de répartition, définie par :

$$F(t, \lambda, \beta) = 1 - e^{-\left(\frac{t}{\lambda}\right)^{\beta}}$$

Finalement, la fonction de survie est alors définie par :

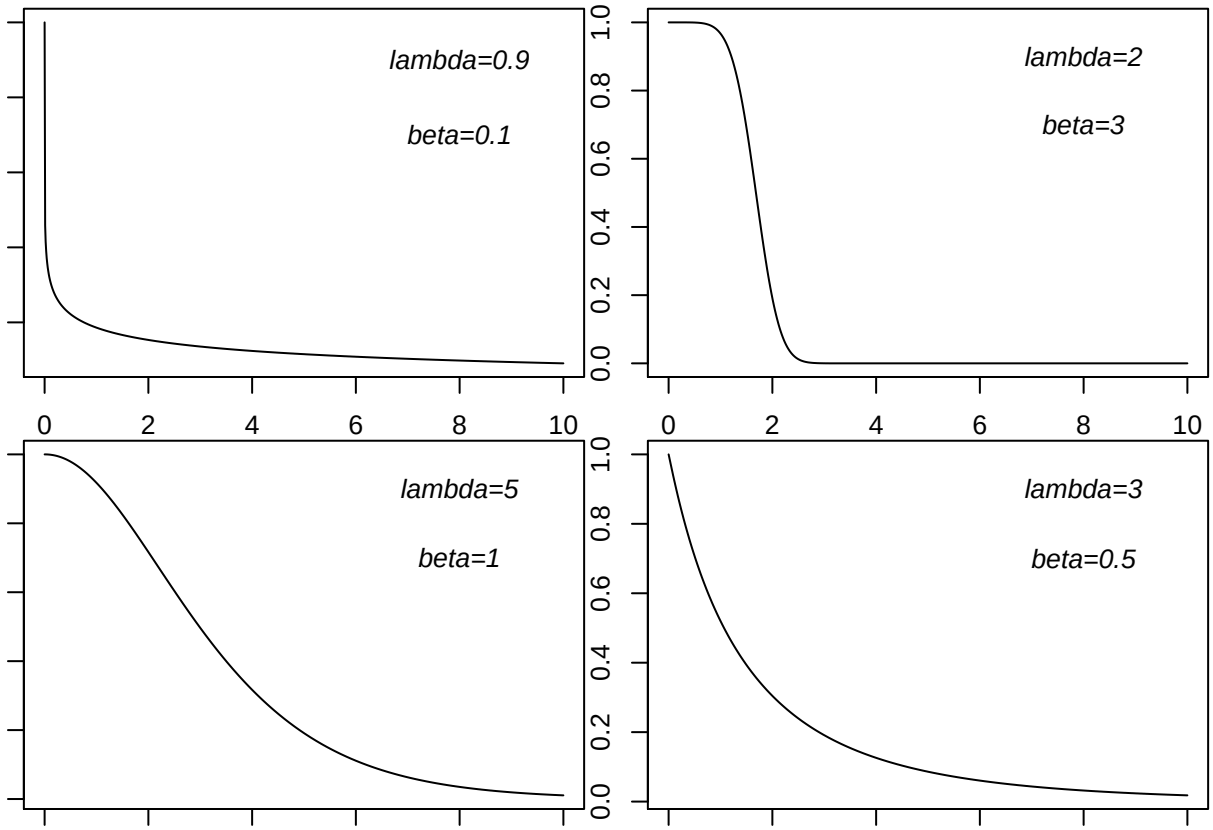
$$S(t, \lambda, \beta) = 1 - F(t, \lambda, \beta) = e^{-\left(\frac{t}{\lambda}\right)^{\beta}}$$

A partir de ces informations, il est possible d'affilier à chaque composante du système, la fonction de survie définie ci-dessus. Ainsi, toutes les composantes de $R_1, ..., R_9$ suivent la même fonction de survie, du moind c'est comme ça qu'on le définit ici. La modélisation n'est pas parfaite mais elle permet d'obtenir les résultats qui suivent.

En implémentant la fonction suivante (trouvable dans l'annexe), on obtient la probabilité de survie du système au bout d'une durée t, selon les paramètre λ et β qu'on prend au choix.

Ensuite, on peut alors pour t variant de 0 à 10 par des pas très faible, tracer un graphe donnant la probabilité de survie du système en fonction du temps, en faisant varier λ et β.

On obtient les résultats suivant :

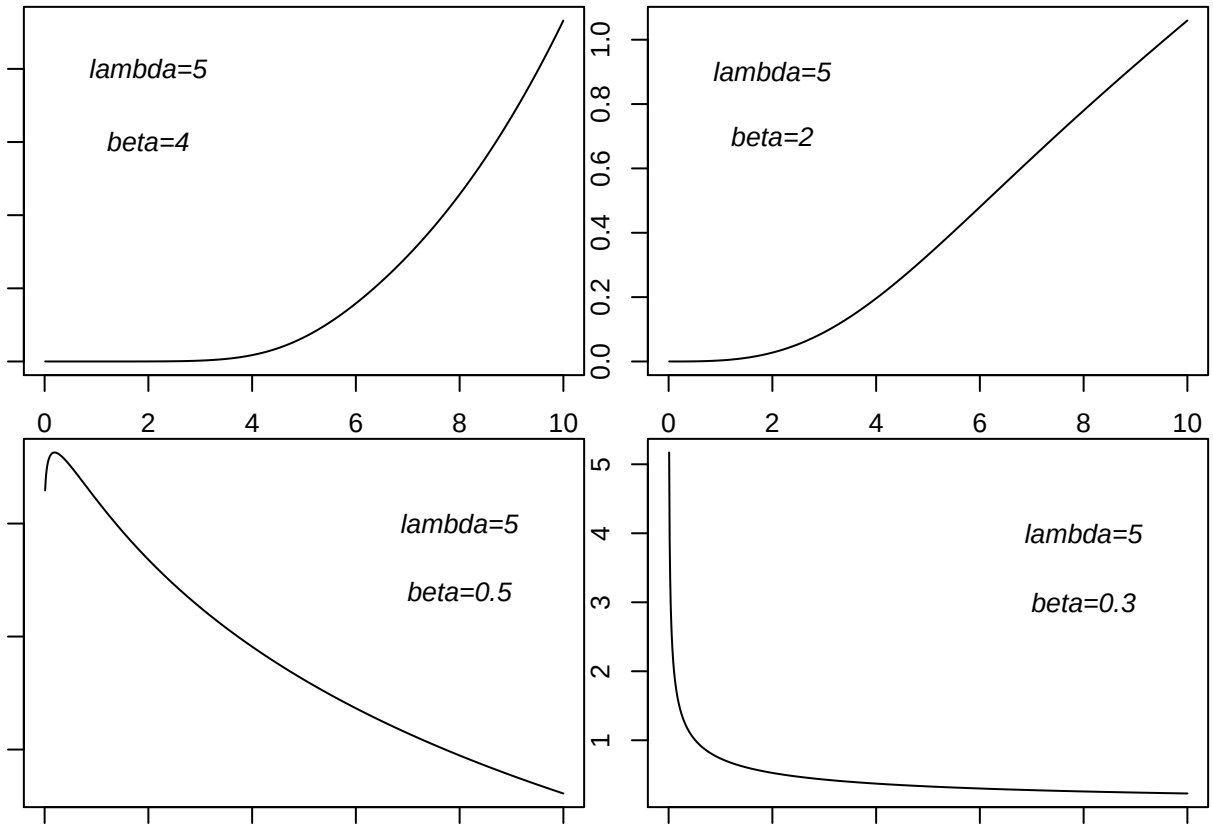


On constate dans un premier temps que toutes les courbes sont décroissantes, ce qui est logique puisque la probabilité que le système tombe en panne augmente au cours du temps, et donc sa probabilité de survie diminue. De plus on constate bien que le paramètre λ correspond à l'instant t sur lequel la courbe se place (paramètre d'échelle), tandis que le paramètre β correspond à la forme de cette courbe; un β petit allonge la durée de vie du système.

2.2.2 Graphes annexes

Il est aussi important de représenter le graphe de la fonction $t \longrightarrow -\frac{ln(\bar{F}(t))}{t}$. Ici, on utilise une fonction R analogue à la précédente ou on change juste à peine la sortie afin d'obtenir la fonction espérée.

En fixant λ à 5, on obtient alors les représentations graphiques suivantes :



Pour les 4 graphes ci-dessus, on remarque que pour $\beta > 1$ la fonction est croissante, tandis que pour $\beta < 1$ la fonction est décroissante.

2.3 Trajectoire des fonctions du système

On sait qu'il est possible de simuler n'importe quelle fonction de répartition G, sachant que G^{-1} est exprimable d'une manière simple. Pour cela, on peut utiliser la loi uniforme. En effet, on a :

$$G^{-1}(u) = inf\{x : G(x) > u, u \in [0,1]\}$$

Alors, $X = G^{-1}(U)$ est une variable aléatoire qui suit une loi de Weibull. La fonction de répartition de cette fonction est donnée par :

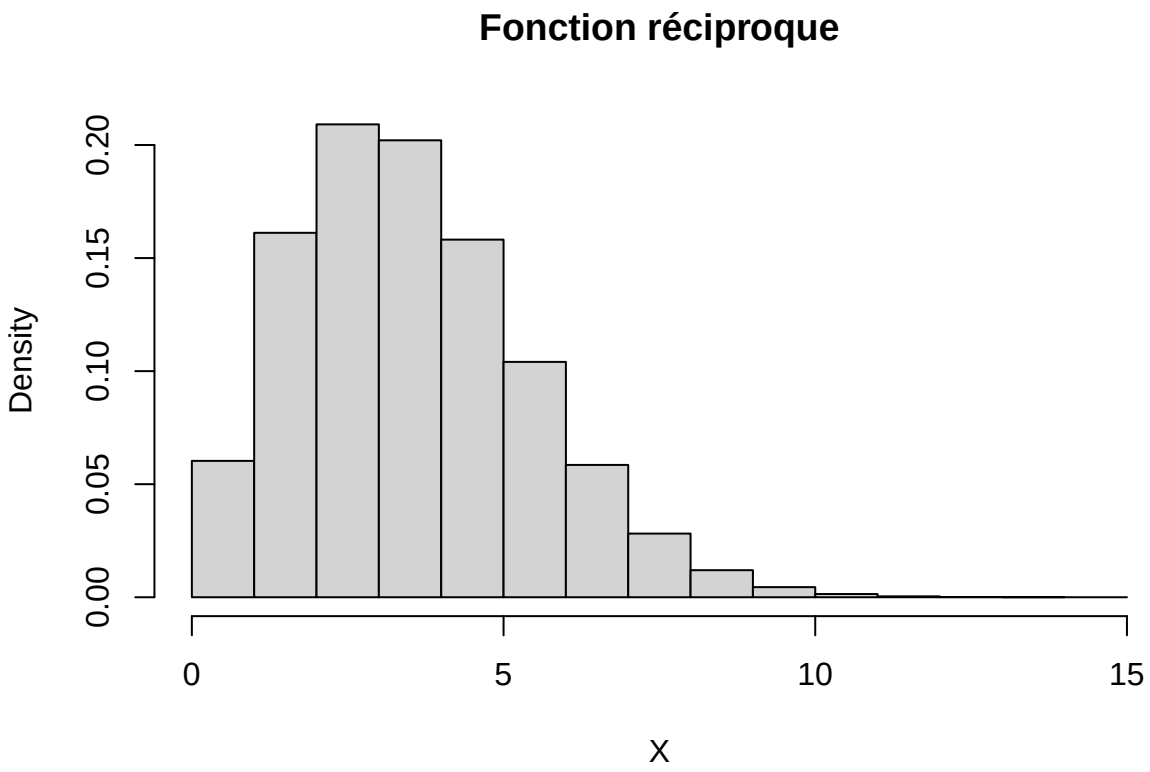
$$G(t, \lambda, \beta) = 1 - e^{-\left(\frac{t}{\lambda}\right)^{\beta}}$$

Et la fonction réciproque est :

$$G^{-1}(u) = \lambda(-ln(1-u))^{\frac{1}{\beta}} = \lambda.(-ln(u))^{\frac{1}{\beta}}$$

car U et 1-U ont la même loi puisque $U \sim Unif([0,1])$.

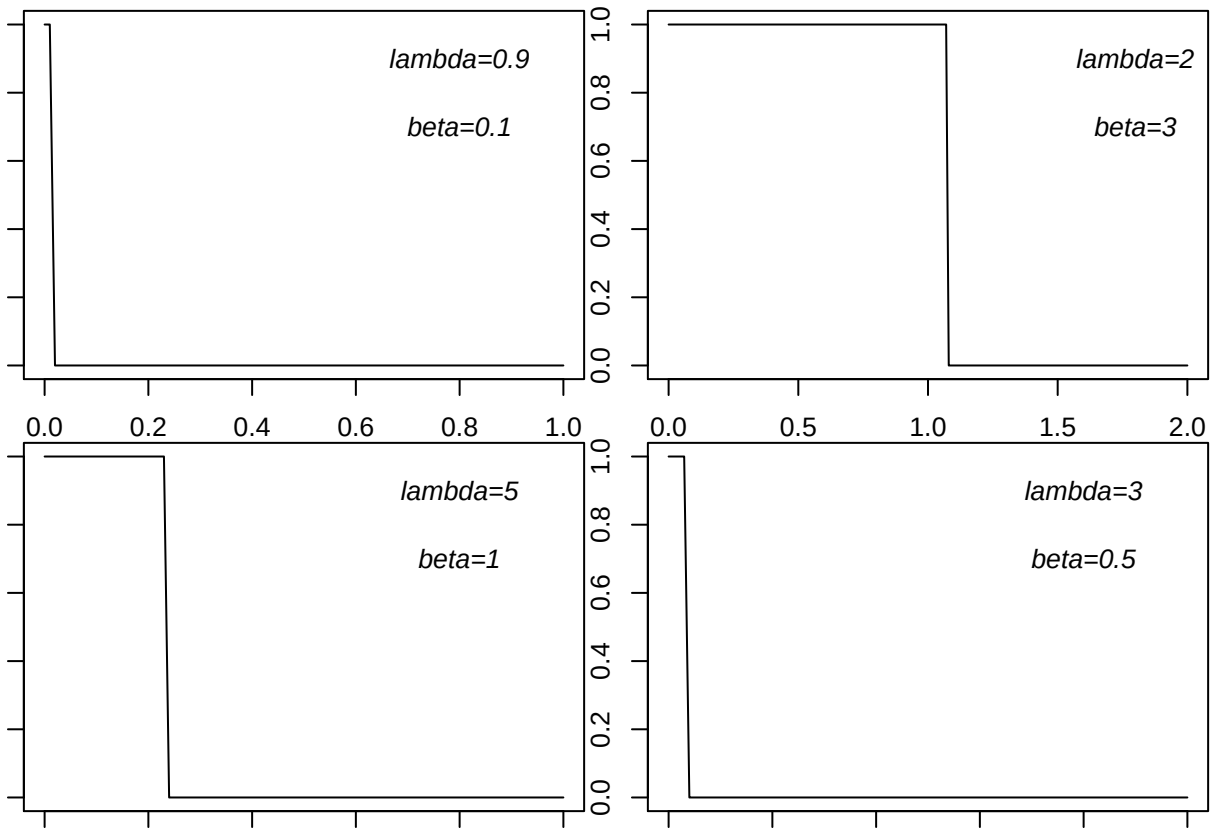
On crée un code qu'il est possible de retrouver dans l'annexe, qui permet de générer les échantillons pour créer la fonction réciproque.



on remarque que l'histogramme de la fonction réciproque est bien semblable à une densité de Weibull. C'est le but de cet exercice puisque ces deux échantillons sont théoriquement les mêmes. On utilise ce résultat pour la suite de la question.

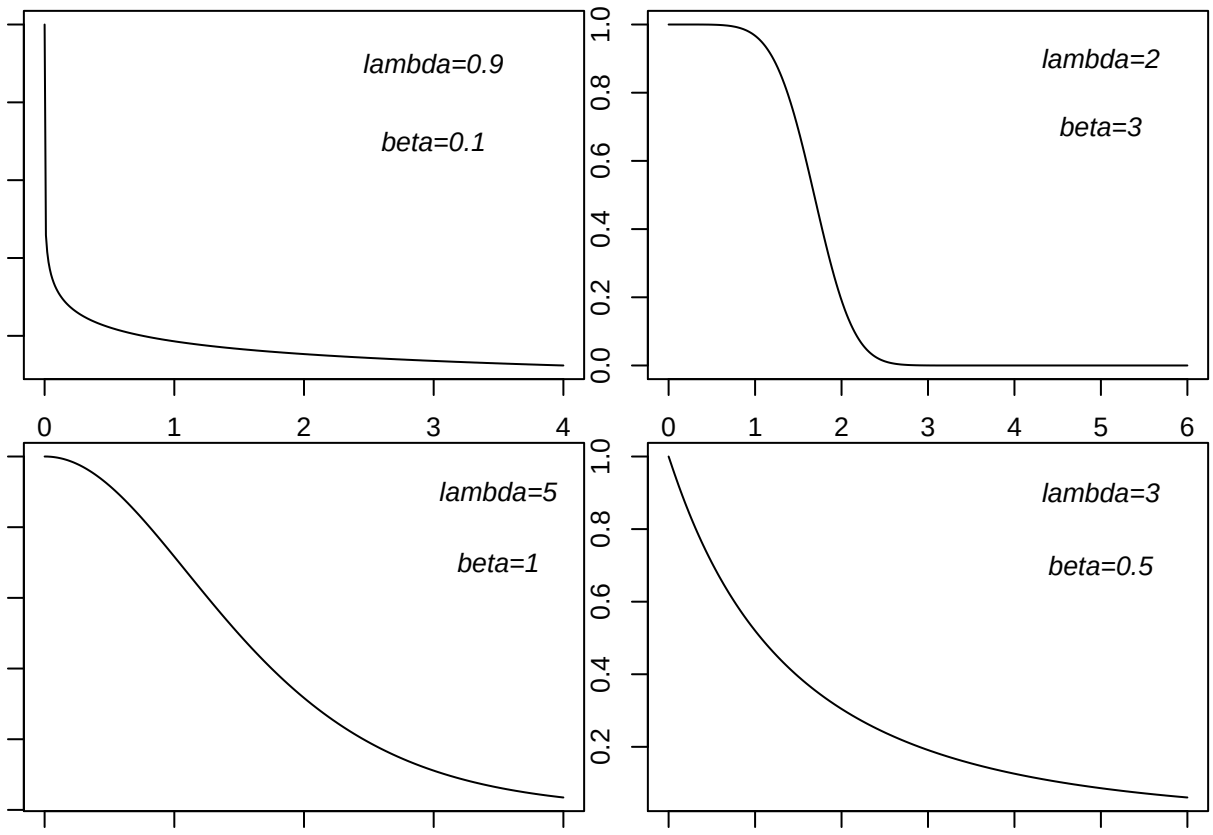
Par la suite, on peut utiliser une fonction afin de tracer quelques graphes représentant une réalisation de $\Phi(X_t)$ sur un intervalle [0,a]. Celle ci est trouvable en annexe.

On trace alors différent graphes sur l'intervalle [0,1], sauf le second qui s'étend un peu plus, et donc pour lequel on prend l'intervalle [0,2] (intervalles petits ici car les durées sont très courtes ici).



On constate donc des fonction qui sont à valeurs dans $\{0, 1\}$.

A partir de cette remarque, il est possible de simuler quelques trajectoires $t \longrightarrow \Phi(X_t)$ sur un intervalle [0,a] avec a un réel qu'on met en entrée, car jusqu'à maintenant ce réel était fixé à 10 subjectivement. On retrouve le code des fonctions légèrement changées dans l'annexe.



Les sorties graphiques nous permettent une meilleure visualisation puisque l'intervalle est plus précis. On constate des courbes relativement cohérentes ici, quoiqu'on remarque de légères différences avec celles produites dans les questions précédentes.

2.4 Estimation de la durée de vie du système

Pour estimer la durée de vie du système, on peut utiliser le fait que :

$$T \simeq t_N, N = max\{n, \Phi(X_{t_i}) = 1\}$$

où $(t_i)_{i=1,\dots,n}$ est une subdivision de pas très petits, et n est grand.

Nous allons donc estimer la durée de vie T du système entier, mais aussi les durées de vie T_1 et T_2 des sous systèmes qui le composent. Pour cela, on réutilise le code précédent en utilisant cette fois ci l'option permettant d'estimer T. Ainsi, on fait le choix de créer un tableau qui représente ces durées de vie pour différentes valeurs de λ et β.

```
##                  Plan Est_T Est_T1 Est_T2
## 1 lambda=2, beta=0.5  0.03   0.15   0.06
## 2   lambda=2, beta=2  0.56   0.70   0.74
## 3 lambda=4, beta=0.5  0.12   0.05   0.19
## 4   lambda=4, beta=2  0.61   0.64   0.68
## 5 lambda=6, beta=0.5  0.02   0.09   0.13
## 6   lambda=6, beta=2  1.05   0.67   0.72
```

On constate des valeurs qui flucutuent énormément dans un premier temps. En effet, le fait de simuler ici cause des problèmes d'instabilité qui dépend de la variance des durées de vie. Une solution utilisant la loi

forte des grands nombres sera abordée dans la partie suivante. On constate ici que T est forcément inférieure à T_1 et T_2, ce qui est assez logique. On note aussi qu'un $\beta < 1$ diminue fortement la durée de vie.

2.5 Espérance de la durée de vie du système

Nous allons donc finalement utiliser la loi des grands nombres pour trouver une espérance approximative de la durée de vie du système. On pose:

$$\mu = E[T], \quad \mu_1 = E[T_1], \quad \mu_2 = E[T_2]$$

Ainsi, nous allons effectuer le code permettant de calculer un nombre N grand de fois, puis nous prendrons la moyenne des valeurs obtenues. Cette méthode permettra donc d'estimer son espérance.

On peut par cette occasion représenter pour différentes valeurs de λ et β.

```
##                   Plan Est_mu Est_mu1 Est_mu2
## 1 lambda=0.5, beta=0.5 0.0711  0.0806  0.1145
## 2   lambda=5, beta=0.5 0.0737  0.0791  0.1235
## 3     lambda=5, beta=2 0.6435  0.7082  0.8551
## 4  lambda=10, beta=0.5 0.0768  0.0785  0.1250
## 5    lambda=10, beta=2 0.6439  0.6906  0.8673
```

Le but pour terminer ce projet est de montrer qu'on a :

$$\mu \geq \left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right)^{-1}$$

seulement lorsque $\beta > 1$, et que lorsque ce n'est pas le cas, de montrer alors qu'il existe un contre-exemple.

Le tableau ci-dessus nous montre par exemple que pour $\lambda = 5$ et $\beta = 2$, on a :

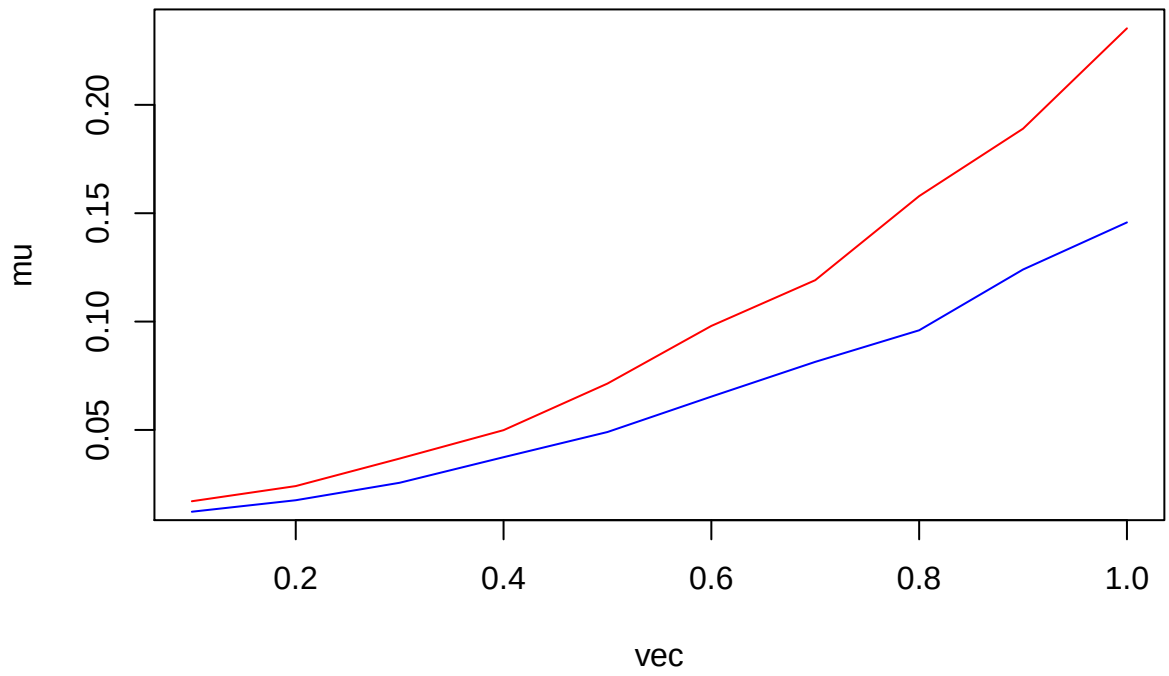
$$\mu = 0.4090 \geq 0.2441047 = \left(\frac{1}{0.4362} + \frac{1}{0.5543}\right)^{-1} = \left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right)^{-1}$$

L'équation est donc bien vérifiée pour ces paramètres. (Attention ! Les valeurs prises ci-dessus (et ci-dessous) ne se retrouvent peut-être pas dans le tableau. Elles sont néanmoins bien réelles.) On remarque aussi que l'égalité est bien valide pour certaines valeurs $\beta < 1$, par exemple, pour $\lambda = 10$ et $\beta = 0.5$:

$$\mu = 0.0586 \geq 0.03908654 = \left(\frac{1}{0.0623} + \frac{1}{0.1049}\right)^{-1} = \left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right)^{-1}$$

Cependant, il a été impossible après plusieurs essais de trouver un cas où l'inégalité n'est pas vérifiée. Mais on remarque tout de même que l'écart entre les deux parties de l'inéquation se rapproche énormément quand $\beta < 1$.

Il est même possible de tracer une graphe où on fixe on fait varier β entre 0.1 et 1 avec un pas de 1, et où on trace en rouge la partie gauche de l'équation, et en bleu la partie droite de l'équation.



On constate que la ligne rouge est toujours au dessus de la ligne bleu. Il faudrait donc encore approfondir pour trouver ce fameux contre exemple... C'est finalement un point d'amélioration notable sur ce compte-rendu.

3 Compte-rendu du TP2

Pour ce second TP, nous allons directement faire la suite du TP précédent puisque les questions abordées concernent le modèle du TP1.

3.1 Disponibilité du système

Dans le cadre de cette partie, les temps de vie des composants suivent des lois de Weibullde paramètre λ et β. On se place dans le cadre où le système est remis à neuf ou réparé, avec inspection éventuelle au bout de T unités de temps si une panne n'a pas eu lieu ou si elle n'a pas été détectée. On note ξ_i la durée entre la (i-1)-ème composante et i-ème remise à neuf du système, et ξ_i^* la i-ème durée d'occupation.

On pose A(t) la disponibilité du système. C'est son aptitude à accomplir une fonction en un temps T donné. On a :

$$A(T) = \frac{E[\xi_i^*]}{E[\xi_i]}$$

Il est aussi possible de définir la disponibilité de la manière suivante :

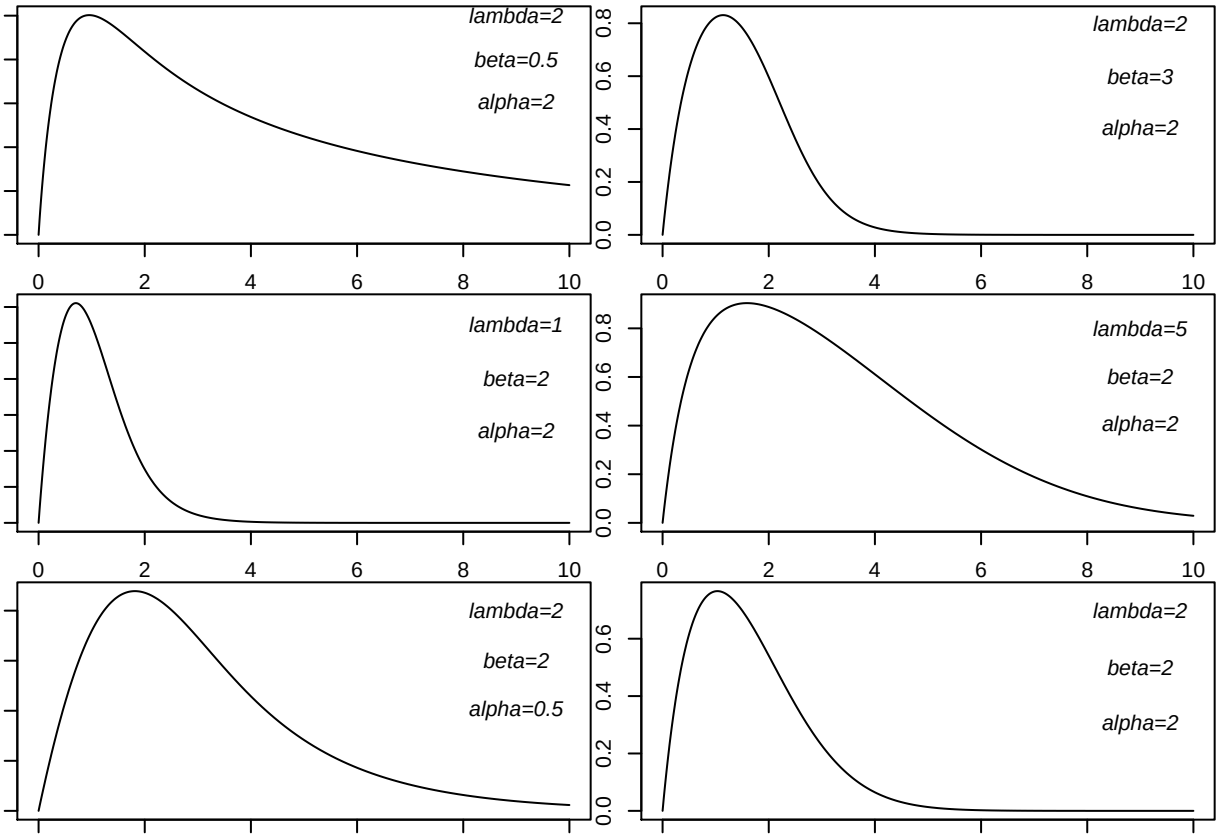
$$A(T) = \int_0^T R(T-u).m(u)du$$

où R(.) est la fiabilité de la loi de Weibull, donnée par $R(t) = e^{-(\frac{t}{\lambda})^\beta}$, et m(.) est la densité de renouvellement, qu'on définit avec la loi exponentielle de paramètre α, ce qui donne $m(u) = \alpha e^{-\alpha u}$. On obtient par conséquent :

$$A(T) = \int_0^T e^{-(\frac{T-u}{\lambda})^\beta}.\alpha e^{-\alpha u}du$$

Ainsi, on implémente le code R tel qu'il est donné dans l'annexe. On notera qu'il utilise la fonction "integrate" de R qui est assez spécifique dans cette utilisation.

Nous allons donc tracer la disponibilité pour différents systèmes, avec des paramètres λ, β, α qui varient. On obtient les graphes suivants :



Ces graphes nous permettent de voir que le paramètre β allonge la disponibilité lorsqu'il devient petit, il rend la distibution à queue épaisse. De son côté, le paramètre λ allonge aussi la distribution, mais lorsqu'il grandit. On constate que la moyenne se situe autour de ce paramètre (empiriquement). Enfin, le paramètre α a aussi un léger impact sur la disponibilité.

3.2 Valeur optimale de la disponibilité

A partir de ces graphes, un détail important peut-être de calculer le T optimal, celui qui maximise A(T). On nomme ce paramètre T^*. On implémente la fonction qui calcule ce T optimal.

Finalement, on effectue le calcul dans le cas $\lambda = \beta = \alpha = 2$. On trouve :

```
## [1] 1.0389610 0.7657403
```

On a donc un T^* qui semble cohérent avec le graphe ci-dessus, qui renvoit donc une probabilité $A(T^*)$ d'un peu plus de 75%. Cela signifie que pour un temps T^*, la probabilité que le système soit opérationnel est maximal, d'environ 75%.

4 Projet

4.1 Fonction de survie du système

Soit un système (S) à n composants de durée de vie indépendantes. Soient $x_1, \cdots, x_n$ l'état des composants avec :

$$\Phi(x_i) = \begin{cases} 1 & si\, le\, composant\, i\, marche \\ 0 & sinon \end{cases}$$

Soit Φ la fonction de structure du système, dite ici binaire :

$$\{0,1\}^n \to \{0,1\}$$

$$\Phi(x_1, \cdots, x_n) = \begin{cases} 1 & si\, le\, système\, fonctionne \\ 0 & sinon \end{cases}$$

Le système complexe que nous allons étudier ici se compose de douze éléments. Son diagramme est le suivant:

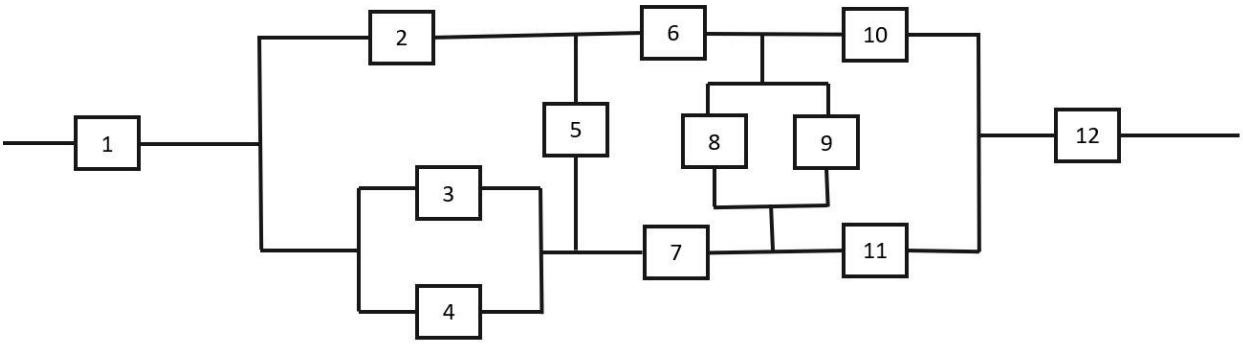


Figure 3: Système représenté

C'est un système pont. On appelle ainsi un système qui ne se réduit pas à une combinaison série-parallèle. Pour déterminer sa fonction de structure, il convient d'utiliser des probabilités conditionnelles : nous allons ainsi considérer plusieurs cas.

- ***Cas où*** x_5***,*** x_8 ***et*** x_9 ***ne fonctionnent pas***

Nous avons le diagramme équivalent ci-dessous:

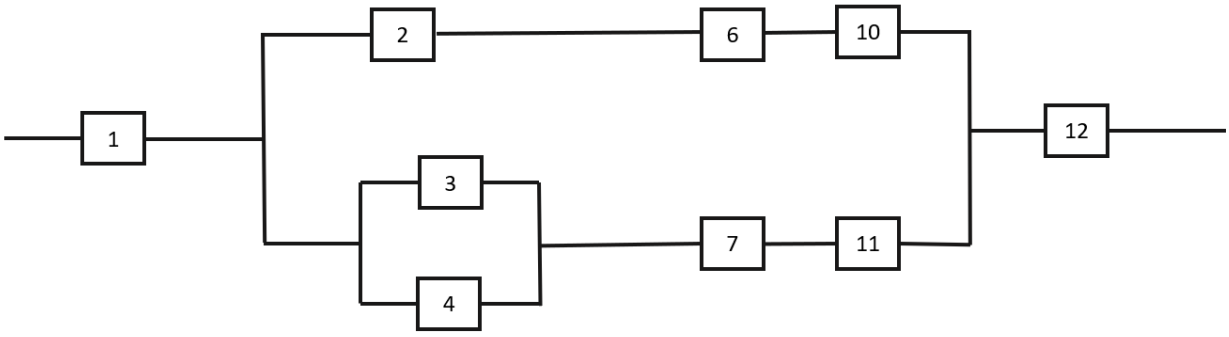


Figure 4: Système représenté 1

Nous avons un système Série/Parallèle avec la fonction de structure suivante:

$$\Phi_1(x_i)_{i=1\cdots12} = x_1\{1-(1-x_2x_6x_{10})(1-x_7x_{11}[1-(1-x_3)(1-x_4])\}x_{12}$$

- ***Cas où*** x_5 ***fonctionne et*** x_8 ***et*** x_9 ***ne fonctionnent pas***

Nous avons le diagramme équivalent ci-dessous:

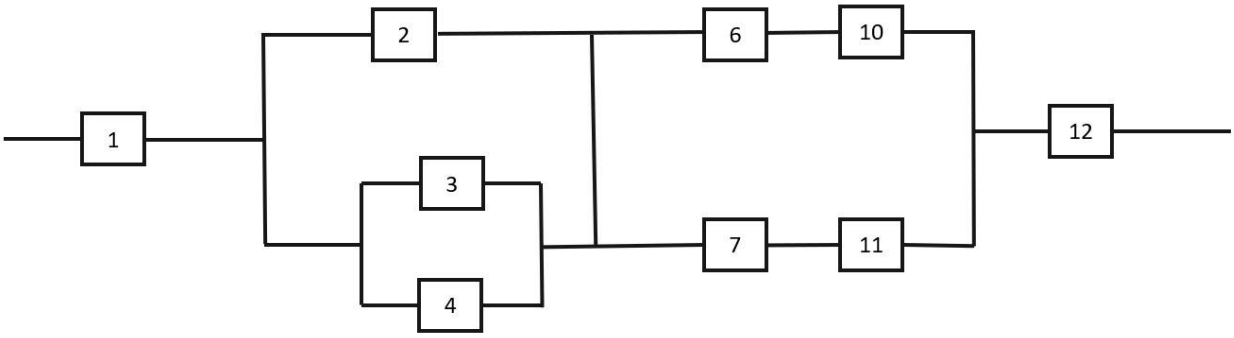


Figure 5: Système représenté 2

Nous avons un système Parallèle/Série que nous pouvons décomposer en deux blocs avec la fonction de structure suivante:

$$\Phi_2(x_i)_{i=1\cdots12} = x_1\{1-(1-x_2)(1-(1-(1-x_3)(1-x_4)))\}\{1-(1-x_6x_{10})(1-x_7x_{11})\}x_{12}$$

- ***Cas où*** x_5 ***ne fonctionne pas et*** x_8 ***ou*** x_9 ***fonctionne***

Nous avons le diagramme équivalent ci-dessous:

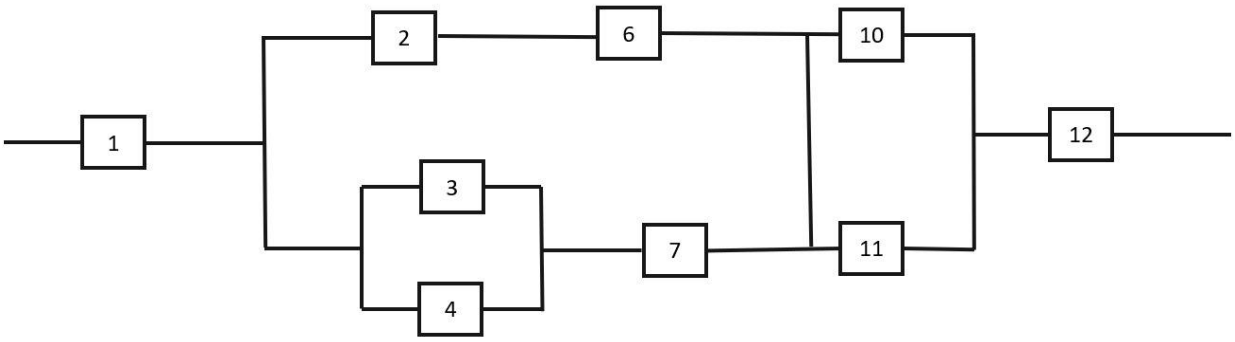


Figure 6: Système représenté 3

Nous avons encore un système Parallèle/Série que nous pouvons décomposer en deux blocs avec la fonction de structure suivante:

$$\Phi_3(x_i)_{i=1\cdots12} = x_1\{1-(1-x_2x_6)(1-x_7(1-(1-x_3)(1-x_4)))\}\{1-(1-x_{10})(1-x_{11})\}x_{12}$$

- ***Cas où*** x_5 ***fonctionne et*** x_8 ***ou*** x_9 ***fonctionne***
  Nous avons le diagramme équivalent ci-dessous:

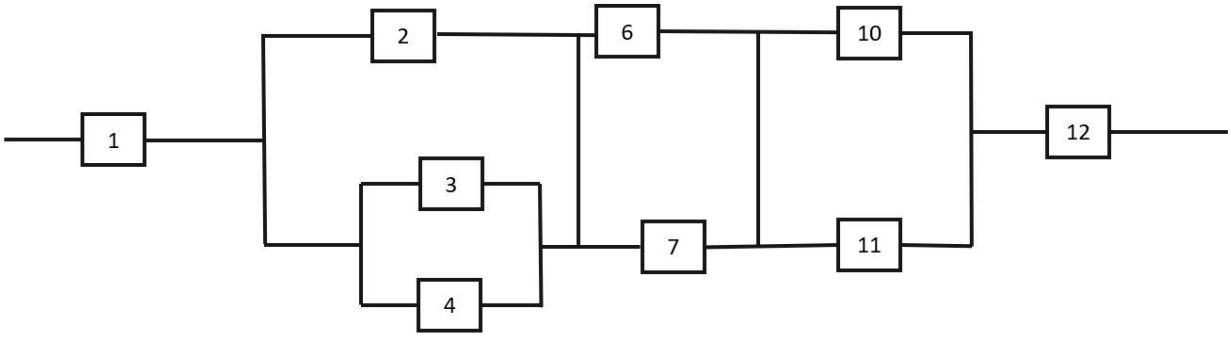


Figure 7: Système représenté 4

Nous avons cette fois-ci un système Parallèle/Série que nous pouvons décomposer en trois blocs avec la fonction de structure suivante:

$$\Phi_4(x_i)_{i=1\cdots12} = x_1\{1-(1-x_2)(1-x_3)(1-x_4)\}\{1-(1-x_6)(1-x_7)\}\{1-(1-x_{10})(1-x_{11})\}x_{12}$$

En posant maintenant $\Phi^{'}(x_5) = x_5$ et $\Phi^{'}(x_8, x_9) = 1-(1-x_8)(1-x_9)$, la fonction de structure de notre système complexe s'écrit comme suit:

$$\Phi(x_i)_{i=1\cdots12} = (1-\Phi^{'}(x_5))(1-\Phi^{'}(x_8, x_9))\Phi_1(x_i)$$

$$+\Phi^{'}(x_5)(1-\Phi^{'}(x_8,x_9))\Phi_2(x_i)+(1-\Phi^{'}(x_5))\Phi^{'}(x_8,x_9)\Phi_3(x_i)+\Phi^{'}(x_5)\Phi^{'}(x_8,x_9)\Phi_4(x_i)$$

Nous avons ci-dessous deux exemples de système. Un premier système qui ne fonctionne pas en bloquant les composants 2,3 et 4 et un second système qui fonctionne pour lequel nous indiquerons le chemin suivi.

```
## [1] 0
```

Le premier renvoie donc logiquement 0...

```
## [1] 1
```

Et le second renvoie 1, signifiant que le système a bien marché, comme le confirme le shéma ci-dessous.

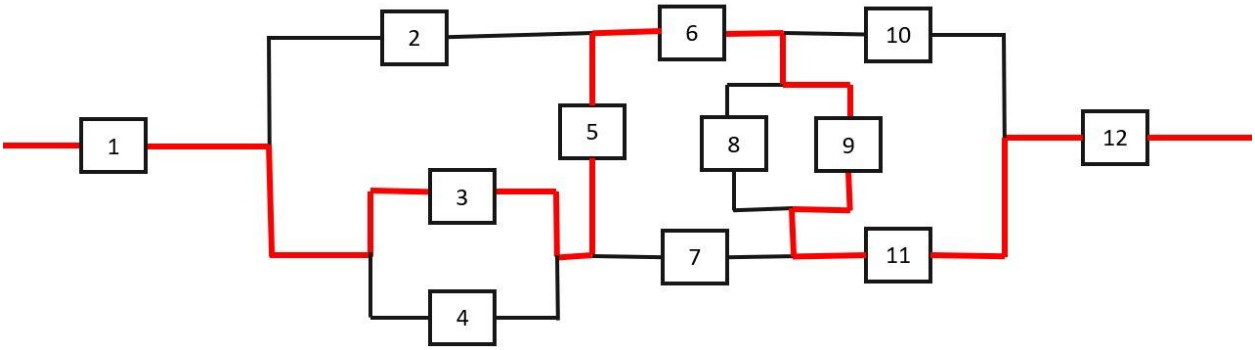


Figure 8: Exemple d'un système qui marche

4.2 Trajectoire et Fonction de survie du système

4.2.1 Fonction donnant la trajectoire

Maintenant que nous avons la fonction de structure de notre système, nous pouvons implémenter une fonction nous donnant une trajectoire du système en entre 0 et T. Pour cela nous allons considérer que les temps de durée de vie sont indépendants et suivent une loi de **Weibull**.

Le code ci-dessous nous permet d'avoir quelques trajectoires de $\Phi(X_t)$ sur un intervalle [0,T] en faisant varier les paramètres λ et β.

4.2.2 Fonction de survie du système

La fonction de survie R_s de notre système s'écrit:

$$R_s(t)=\Phi(R_1,R_2,\cdots,R_{12})$$

On obtient donc:

$$R_s(t)=(1-R_5)(1-R_8)(1-R_9)(R_1\{1-(1-R_2R_6R_{10})(1-R_7R_{11}(R_3)$$
$$+R_4-R_3R_4))\}R_{12})+R_5(1-R_8)(1-R_9)(R_1\{1-(1-R_2)(1-R_3)(1-R_4)\}\{1-(1-R_6R_{10})(1-R_7R_{11})\}R_{12})$$

$$+(1-R_5)(1-(1-R_8)(1-R_9))(R_1\{1-(1-R_2R_6)(1-R_7(R_3+R_4-R_3R_4))\}\{R_{10}+R_{11}-R_{10}R_{11}\}R_{12})$$
$$+R_5(1-(1-R_8)(1-R_9))(R_1\{1-(1-R_2)(1-R_3)(1-R_4)\}\{1-(1-R_6)(1-R_7)\}\{1-(1-R_{10})(1-R_{11})\}R_{12})$$

En implémentant la fonction suivante, on obtient une estimation de la probabilité de survie du système au bout d'une durée t, selon les paramètre λ et β qu'on prend au choix.

En fixant $\lambda = 5$ et $\beta = 0.4$ on obtient respectivement les probabilité suivantes pour t = 2.5, 1à, et 30 :

```
## [1] 0.07903954
```

```
## [1] 0.005546997
```

```
## [1] 0.0001362393
```

On remarque bien que plus on augmente le paramètre t, plus l'estimation de la probabilité de survie diminue. ce qui est bien normal car la fonction de survie est décroissante.

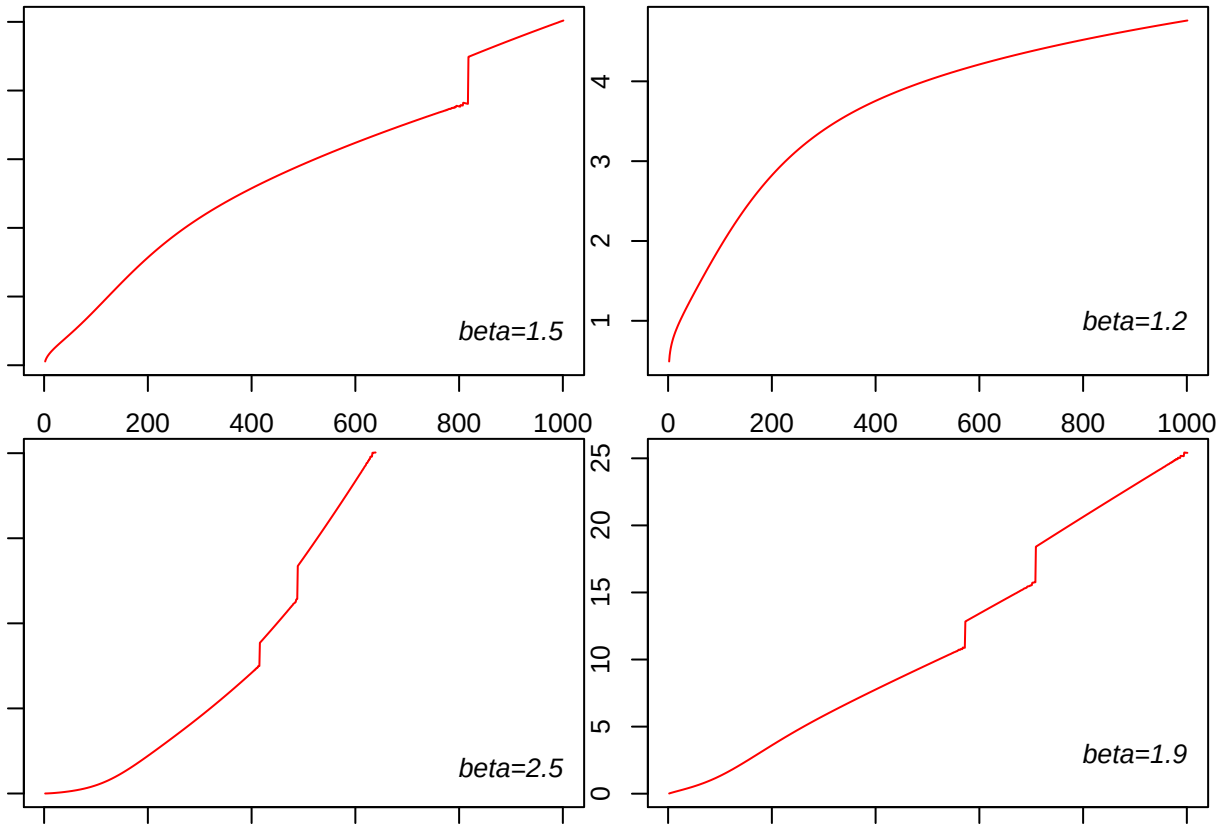
4.3 Système IFRA

Une durée de vie T de **défaillance** λ est dite *IFRA* si elle vérifie la propriété suivante: La fonction qui à $t > 0$ associe: $\frac{1}{t}\int_0^t \lambda(s)\mathrm{d}s$ est croissante en $t \iff$ la fonction $-\frac{\log(R_t)}{t}$ est croissante en t.

En considérant le cas où tous les composants sont IFRA, nous allons vérifier que notre système entier est lui aussi IFRA grâce à une fonction que nous allons implémenter et qui va nous renvoyer des résulats sur un intervalle donné:

```
##  [1] 0.4906616 0.5654074 0.6153171 0.6541944 0.6867309 0.7151277 0.7406050
##  [8] 0.7639131 0.7855483 0.8058562 0.8250878 0.8434311 0.8610306 0.8780000
```

On a bien des valeurs de plus en plus croissantes. Ce qui justifie ainsi que notre système complexe est, lui aussi, IFRA. Représentons maintenant cette fonction pour des valeurs de $\beta > 1$ (période de vieillissement selon la courbe en baignoire) et pour $\alpha = 1.5$ fixé:

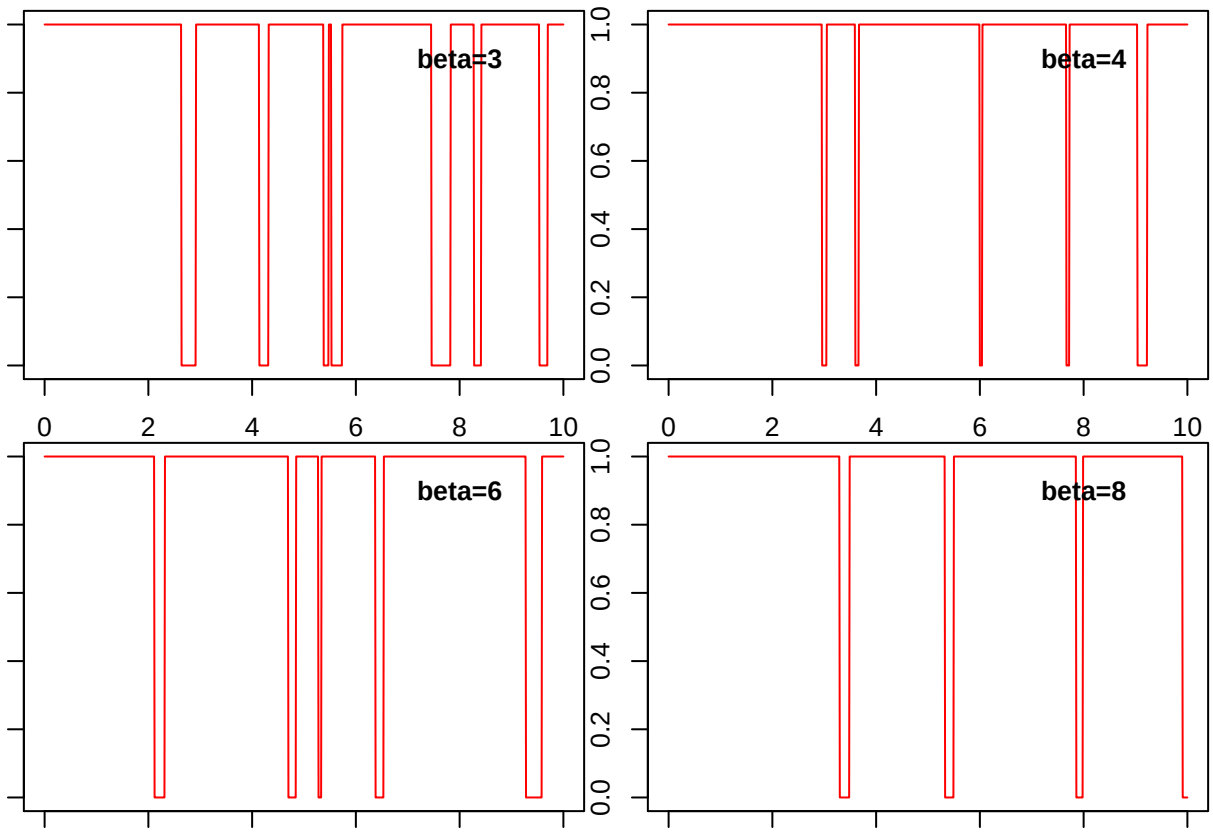


4.4 En cas de réparation : état du système

Ici, on se place dans le cas où des réparations sont possibles pour chaque composant. Les durées de vie (T_i) et de réparation (R_i) sont i.i.d, indépendantes et de lois de Weibull pour la durée de vie, tandis qu'on choisit de faire suivre une loi exponentielle pour les durées de réparation.

Alors on implémente le code R tel qu'il est donné dans l'annexe afin d'avoir l'état du système sur un intervalle $[0, w]$. Notons que le code utilisé est particulièrement complexe et ingénieux à la fois avec l'utilisation de la récursivité. Il vaut le coup d'oeil !

Ainsi, nous pouvons simuler et tracer quelques trajectoires de $\Phi(X_t)$. On obtient les trajectoires suivantes :



On remarque des trajectoires qui varient entre 0 et 1. On a choisi de fixer $\alpha = 0.01$, signifiant un temps de réparation assez long, et $\lambda = 10$. Finalement, on remarque que le temps de panne est plus grand quand β est faible.

4.5 Nombre de pannes, Temps de fonctionnement

4.5.1 Création des fonctions

Le nombre de pannes attendu (Expected Number of Failure) que nous allons noté N sur un intervalle $[a, b]$ se définit comme :

$$N[a, b] = \int_a^b f(t)\mathrm{d}t = \int_a^b h(t)R(t)\mathrm{d}t$$

où $h(t)$ représente le taux de défaillance et $R(t)$ la fonction de survie du système.Le système fonctionne jusqu'à ce qu'il y'ait une défaillance. Le temps de fonctionnement Y_w comme le temps moyen de fonctionnement jusqu'à la prochaine panne. Celui-ci se définit par:

$$MTTF = \int_0^{+\infty} tf(t)\mathrm{d}t = \int_0^{+\infty} R(t)\mathrm{d}t$$

On crée donc une fonction qui calcule le nombre de pannes.

En voici quelques exemples :

Pour w=10, $\lambda = 5, \beta = 3, \alpha = 0.01$, on a un nombre de panne de :

```
## [1] 12
```

Pour w=10, $\lambda = 10, \beta = 2, \alpha = 1$, on a un nombre de panne de :

```
## [1] 16
```

Pour w=10, $\lambda = 1, \beta = 8, \alpha = 0.01$, on a un nombre de panne de :

```
## [1] 20
```

Pour w=10, $\lambda = 10, \beta = 8, \alpha = 0.01$, on a un nombre de panne de :

```
## [1] 3
```

On peut par la suite créer une fonction qui calcule cette fois Y_w, le temps de fonctionnement sur cet intervalle.
En voici quelques exemples, pour les mêmes valeurs qu'avant :
Pour w=10, $\lambda = 5, \beta = 3, \alpha = 0.01$, on a un nombre de panne de :

```
## [1] 8.07
```

Pour w=10, $\lambda = 10, \beta = 2, \alpha = 1$, on a un nombre de panne de :

```
## [1] 9.64
```

Pour w=10, $\lambda = 1, \beta = 8, \alpha = 0.01$, on a un nombre de panne de :

```
## [1] 6.85
```

Pour w=10, $\lambda = 10, \beta = 8, \alpha = 0.01$, on a un nombre de panne de :

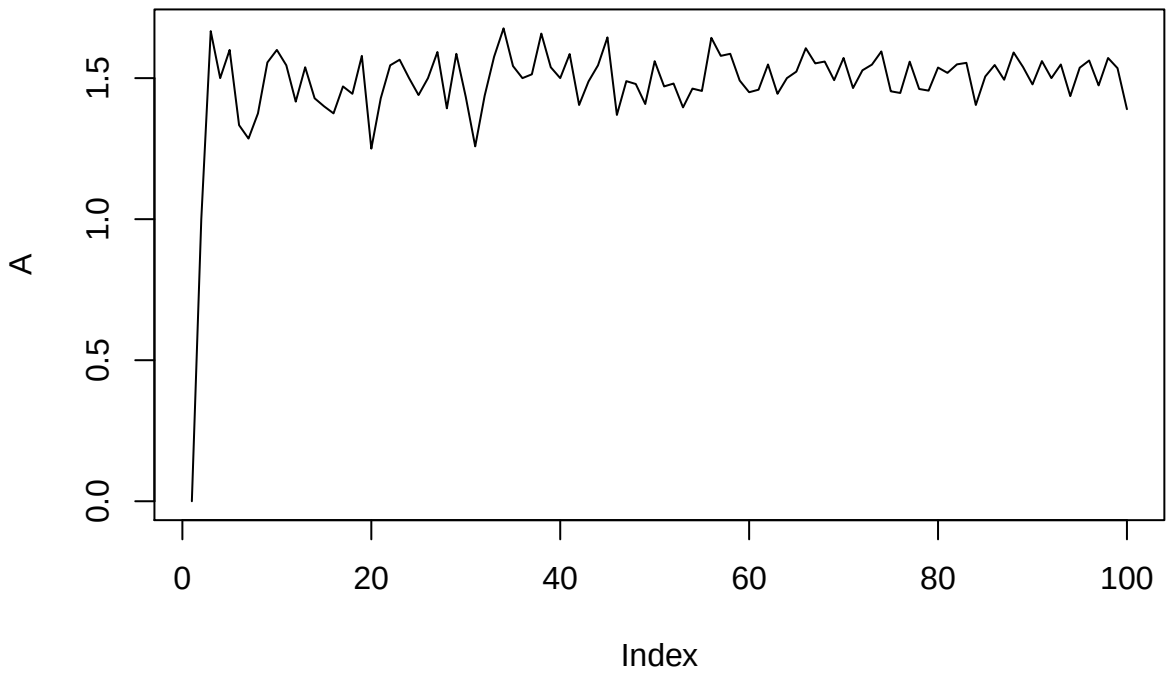
```
## [1] 9.65
```

4.5.2 Loi forte des grands nombres, TCL

Nous allons maintenant vérifier si la formule :

$$\frac{N[0,w]}{w} \longrightarrow m$$

est plausible quand w tend vers $+\infty$.

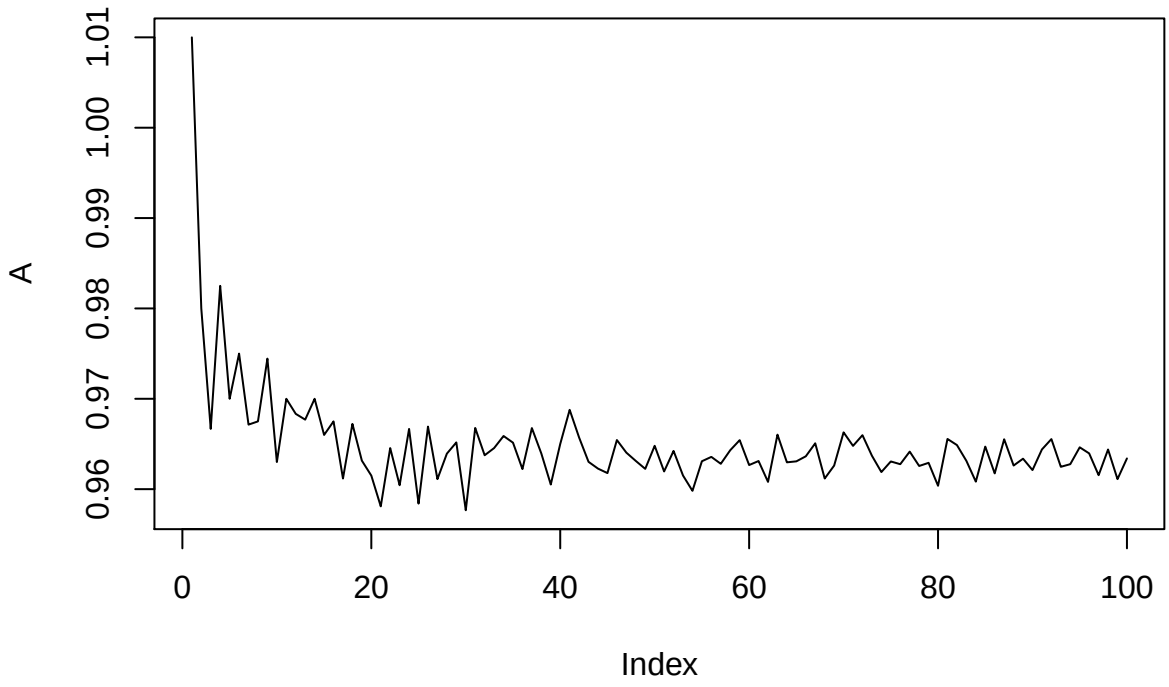


On constate ici que lorsque w augmente (on le fait aller jusqu'à 100), la quantité N[0,w]/w semble tendre vers une valeur précise.. En fixant empiriquement cette valeur à m=1.5, on conclus donc la validité de cette proposition.

On vérifie ensuite si la formule :

$$\frac{Y_w}{w} \longrightarrow l$$

est plausible quand w tend vers $+\infty$.

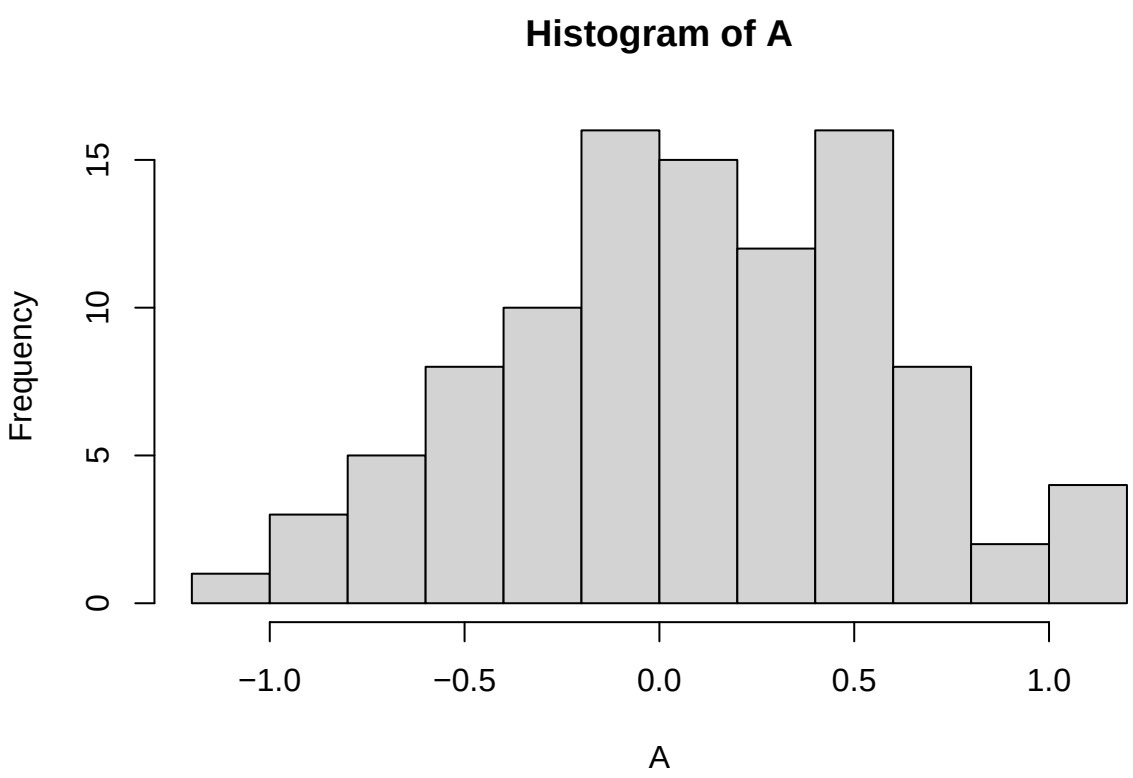


Ici encore, la valeur semble converger quand w grandit, vers une valeur l=0.962 (choisie empiriquement) on conclut donc la validité de cette proposition.

Enfin, il s'agit de démontrer que :

$$\frac{N[0,w] - mw}{\sqrt{w}} \longrightarrow N(0,\sigma^2)$$

quand $w \longrightarrow +\infty$.



Ici, la loi normale n'est pas énormément visible non plus, mais on constate bien une loi centrée en 0.. On peut supposer qu'en augmentant encore la valeur de w, cette loi sera d'autant plus visible...

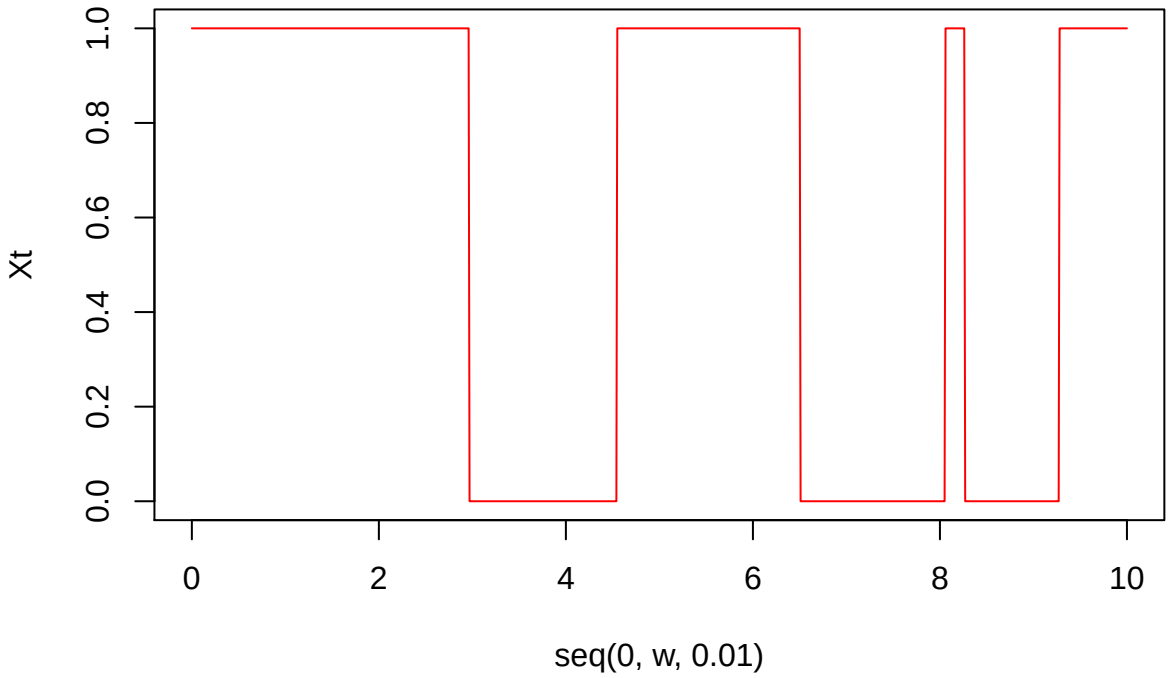
4.6 Mean Time to failure (MTTF) après maintenance du système

On suppose ici que chaque composant est soumis à maintenance, plutôy que réparation. Autrement dit, nous décidons que le système est remis en état au bout d'un temps fixe T, et que le temps de remise à neuf dure δ pendant lequel le système n'est pas opérationnel. Après la remise à neuf, tous les composants sont opérationnels.

Nous allons comparer le MMTF sur un intervalle fixé grand, en prenant des paramètres tels que les deux solution sembleront être équivalentes.

Pour cela, on peut reprendre la fonction utilisée tout à l'heure, en la changeant légerement de manière à respecter cet énoncé.

On peut créer des graphes où on a les mêmes paramètres $\lambda = 10$ et $\beta = 4$, et ou on prend une durée T de 1:



On peu comparer ce graphe à un modèle ou le temps de réparartion est modélisé par une loi exponentielle de paramètre 0.00021 environ. Ainsi, les résultats obtenus seront cohérents.

5 Conclusion

En définitive, ces travaux présentent des contributions aux notions apprises en cours. Les principaux objectifs étaient de reproduire des simulations des trajectoires des systèmes étudiés, d'implémenter des fonctions de fiabilité.

Dans le TP1, nous avons étudié un système constitué de deux blocs en série pour lesquels nous avons implémeter les fonctions de structure, de survie, de répartition lorsque des durées de vie suivent des lois de Weibull, le taux de défaillance et nous avons estimé les durées de vie du système et les durées de vie moyennes des deux blocs.

Dans le TP2, nous avons étudié deux systèmes dont le premier a des durées de vie et de réparation qui suivent des lois exponentielles et pour le second des lois de Weibull. Les fonctions implémentées sont les mêmes.

Dans le projet nous avons étudié un système pont constitué de douze composants.Nous avons implémeter les fonctions de structure, de survie, de répartition lorsque des durées de vie suivent des lois de Weibull, le taux de défaillance et nous avons estimé les durées de vie du système et les durées de vie moyennes des deux blocs pour des durées de vie qui suivent des lois de Weibull..

Ce projet a été, pour nous, très enrichissant car il nous a permis d'évaluer la fiabilité par des techniques algorithmiques lorsque la configuration du système est trop compliquée ou assez simple.

6 Annexe

6.1 Code TP1

```
## Question 1

phi <- function(x, sys=0){
  if(sys==1){
    phi1 <- x[3]*((1-(1-x[1])*(1-x[2]))*(1-(1-x[4])*(1-x[5]))) +
      (1-x[3])*(1-(1-x[1]*x[4])*(1-x[2]*x[5]))
    res <- phi1
  }else if (sys==2){
    phi2 <- 1-(1-x[1]*x[2])*(1-x[3])*(1-x[4])
    res <- phi2
  }else{
    phi1 <- x[3]*((1-(1-x[1])*(1-x[2]))*(1-(1-x[4])*(1-x[5]))) +
      (1-x[3])*(1-(1-x[1]*x[4])*(1-x[2]*x[5]))
    phi2 <- 1-(1-x[6]*x[7])*(1-x[8])*(1-x[9])
    res <- phi1*phi2
  }
  res
}

phi(c(1,1,1,0,0,1,1,0,1))
phi(c(1,0,1,0,1,1,1,0,0))

## Question 2

tpSurvSys <- function(t, lambda, beta, simu=F){
  if(simu){
    coeff <- 1 - pweibull(t, beta, lambda)
  }
  else{
    coeff <- exp(-(t/lambda)^beta)
  }
  R <- rep(coeff,9)
  phi(R)
}

## Question 3

vectSys <- function(lambda, beta, a=10, ln=F, simu=F){
  vec <- seq(0, a, 0.01)
  k=1
  for(i in seq(0, a, 0.01)){
    vec[k] <- tpSurvSys(i, lambda, beta, simu)
    if(ln){
      vec[k] <- -log(vec[k])/i
    }
    k=k+1
  }
  plot(seq(0, a, 0.01), vec, type="l", xlab="durée t", ylab="y")
}
```

```r
par(mfrow = c(2,2), mar = c(1,1,1,1))
vectSys(0.9,0.1)
text(8,0.9,"lambda=0.9",font=3)
text(8,0.7,"beta=0.1",font=3)
vectSys(2,3)
text(8,0.9,"lambda=2",font=3)
text(8,0.7,"beta=3",font=3)
vectSys(5,1)
text(8,0.9,"lambda=5",font=3)
text(8,0.7,"beta=1",font=3)
vectSys(3,0.5)
text(8,0.9,"lambda=3",font=3)
text(8,0.7,"beta=0.5",font=3)

## Question 4

par(mfrow = c(2,2), mar = c(1,1,1,1))
vectSys(5,4, ln=T)
text(2,4,"lambda=5",font=3)
text(2,3,"beta=4",font=3)
vectSys(5,2, ln=T)
text(2,0.9,"lambda=5",font=3)
text(2,0.7,"beta=2",font=3)
vectSys(5,0.5, ln=T)
text(8,0.4,"lambda=5",font=3)
text(8,0.37,"beta=0.5",font=3)
vectSys(5,0.3, ln=T)
text(8,4,"lambda=5",font=3)
text(8,3,"beta=0.3",font=3)

## Question 5

n <- 1000000
lambda <- 4
beta <- 2
a <- seq(0, 10, by = 10/(n-1))
X <- rep(0, n)
u <- runif(n,0,1)
X <- lambda*((-log(u))^(1/beta))
hist(X, main="Fonction réciproque", freq=F)

estDVS <- function(lamda, beta, a=10, sys=0, est=T){
  if(sys==1){
    k <- 5
  }else if (sys==2){
    k <- 4
  }else{
    k <- 9
  }
  t <- seq(0, a, 0.01)
  s <- length(t)
  R <- matrix (rep(0, s*k), s, k)
  S <- rep(0, s)
```

```
  for(i in 1:s){
    y <- exp(-(t[i]/lambda)^beta)
    R[i,] <- rep(y, k)
    S[i] <- phi(R[i,], sys)
  }
  X <- matrix(rep(1, s*k), s, k)
  u <- matrix(rep(0, s*k), s, k)
  for(i in 2:s){
    for(j in 1:k){
      u[i,j] <- runif(1,0,1)
      if(X[i-1,j]==1){
        if(u[i,j]>R[i,j]){ X[i,j] <- 0 }
        else{ X[i,j] <- 1 }
      }
      if(X[i-1,j]==0){ X[i,j] <- 0 }
    }
  }
  Xt <- rep(1,s)
  for(i in 1:s){
    Xt[i] <- phi(X[i,], sys)
  }
  if(est){
    sum(Xt)/100
  }
  else{
    plot(t, Xt, type="l")
  }
}

par(mfrow = c(2,2), mar = c(1,1,1,1))
estDVS(0.9, 0.1, a=1, est=F)
text(0.8,0.9,"lambda=0.9",font=3)
text(0.8,0.7,"beta=0.1",font=3)
estDVS(2, 3, a=2, est=F)
text(1.8,0.9,"lambda=2",font=3)
text(1.8,0.7,"beta=3",font=3)
estDVS(5, 1, a=1, est=F)
text(0.8,0.9,"lambda=5",font=3)
text(0.8,0.7,"beta=1",font=3)
estDVS(3, 0.5, a=1, est=F)
text(0.8,0.9,"lambda=3",font=3)
text(0.8,0.7,"beta=0.5",font=3)

par(mfrow = c(2,2), mar = c(1,1,1,1))
vectSys(0.9, 0.1, a=4, simu=T)
text(3,0.9,"lambda=0.9",font=3)
text(3,0.7,"beta=0.1",font=3)
vectSys(2, 3, a=6, simu=T)
text(5,0.9,"lambda=2",font=3)
text(5,0.7,"beta=3",font=3)
vectSys(5, 1, a=8, simu=T)
text(7,0.9,"lambda=5",font=3)
text(7,0.7,"beta=1",font=3)
```

```r
vectSys(3, 0.5, a=6, simu=T)
text(5,0.9,"lambda=3",font=3)
text(5,0.7,"beta=0.5",font=3)

## Question 6

data.frame(Plan = c("lambda=2, beta=0.5", "lambda=2, beta=2", "lambda=4, beta=0.5",
                    "lambda=4, beta=2", "lambda=6, beta=0.5", "lambda=6, beta=2"),
                 Est_T = c(estDVS(2, 0.5), estDVS(2, 2), estDVS(4, 0.5),
                           estDVS(4, 2), estDVS(6, 0.5), estDVS(6, 2)),
                 Est_T1 = c(estDVS(2, 0.5, sys=1), estDVS(2, 2, sys=1),
                            estDVS(4, 0.5, sys=1), estDVS(4, 2, sys=1),
                            estDVS(6, 0.5, sys=1), estDVS(6, 2, sys=1)),
                 Est_T2 = c(estDVS(2, 0.5, sys=2), estDVS(2, 2, sys=2),
                            estDVS(4, 0.5, sys=2), estDVS(4, 2, sys=2),
                            estDVS(6, 0.5, sys=2), estDVS(6, 2, sys=2)))

## Question 7

estDVS2 <- function(lamda, beta, a=10, sys=0){
  B <- 1:100
  for(i in 1:100){
    B[i] <- estDVS(lambda, beta, a, sys)
  }
  mean(B)
}

set.seed(1)
data.frame(Plan = c("lambda=0.5, beta=0.5", "lambda=5, beta=0.5",
                    "lambda=5, beta=2", "lambda=10, beta=0.5","lambda=10, beta=2"),
                 Est_mu = c(estDVS2(0.5, 0.5), estDVS2(5, 0.5), estDVS2(5, 2),
                            estDVS2(10, 0.5), estDVS2(10, 2)),
                 Est_mu1 = c(estDVS2(0.5, 0.5, sys=1), estDVS2(5, 0.5, sys=1),
                             estDVS2(5, 2, sys=1), estDVS2(10, 0.5, sys=1),
                             estDVS2(10, 2, sys=1)),
                 Est_mu2 = c(estDVS2(0.5, 0.5, sys=2), estDVS2(5, 0.5, sys=2),
                             estDVS2(5, 2, sys=2), estDVS2(10, 0.5, sys=2),
                             estDVS2(10, 2, sys=2)))

vec <- seq(0.1, 1, 0.1)
mu <- seq(0.1, 1, 0.1)
test <- seq(0.1, 1, 0.1)
for(i in 1:10){
  mu[i] <- estDVS2(0.005, vec[i])
  test[i] <- (1/estDVS2(0.005, vec[i], sys=1) + 1/estDVS2(0.05, vec[i], sys=2))^(-1)
}
plot(mu~vec, type="l", col="red")
lines(test~vec, col="blue")
```

6.2 Code TP2

```
## Question 1

Avai <- function(lambda, beta, alpha, plot=T){
  A <- array();
  x <- array();
  i <- 1;
  for(t in seq(0, 10, length.out=1001)){
    fun <- function(x){
      Frep <- pweibull(t-x, beta, lambda);
      m <- alpha*exp(-alpha*x);
      R  <- 1-Frep;
      return(R*m);
    }
    A[i] <- integrate(fun, 0, t)$value;
    x[i] <- t;
    i <- i+1;
  }
  if(plot){
    plot(x, A, type="l")
  }else{
    A
  }
}

par(mfrow = c(3,2), mar = c(1,1,1,1))

Avai(2, 0.5, 2)
text(9,0.5,"lambda=2",font=3)
text(9,0.4,"beta=0.5",font=3)
text(9,0.3,"alpha=2",font=3)

Avai(2, 3, 2)
text(9,0.8,"lambda=2",font=3)
text(9,0.6,"beta=3",font=3)
text(9,0.4,"alpha=2",font=3)

Avai(1, 2, 2)
text(9,0.55,"lambda=1",font=3)
text(9,0.4,"beta=2",font=3)
text(9,0.25,"alpha=2",font=3)

Avai(5, 2, 2)
text(9,0.8,"lambda=5",font=3)
text(9,0.6,"beta=2",font=3)
text(9,0.4,"alpha=2",font=3)

Avai(2, 2, 0.5)
text(9,0.4,"lambda=2",font=3)
text(9,0.3,"beta=2",font=3)
text(9,0.2,"alpha=0.5",font=3)
```

```
Avai(2, 2, 2)
text(9,0.7,"lambda=2",font=3)
text(9,0.5,"beta=2",font=3)
text(9,0.3,"alpha=2",font=3)

## Question 2

optAvai <- function(lambda, beta, alpha){
  A <- Avai(lambda, beta, alpha, plot=F)
  for(i in 1:1001){
    if(A[i]==max(A)){
      T <- c(i*10/1001, A[i])
    }
  }
  T
}

optAvai(2,2,2)
```

6.3 Code Projet

```
 ## Question 1
  ## Cas1
     phi1<-function(x){
  y=x[1]*(1-(1-x[2]*x[6]*x[10])*(1-x[7]*x[11]*(x[3]+x[4]-x[3]*x[4])))*x[12]
  return(y)
     }
  ## Cas2
phi2<-function(x){
  y=x[1]*(1-(1-x[2])*(1-x[3])*(1-x[4]))*(1-(1-x[6]*x[10])*(1-x[7]*x[11]))*x[12]
  return(y)
}

  ## Cas3
phi3<-function(x){
  y=x[1]*(1-(1-x[2]*x[6])*(1-x[7]*(x[3]+x[4]-x[3]*x[4])))*(x[10]+x[11]-x[10]*x[11])*x[12]
  return(y)
}
  ## Cas 4
phi4<-function(x){
  y=x[1]*(1-(1-x[2])*(1-x[3])*(1-x[4]))*(1-(1-x[6])*(1-x[7]))*(1-(1-x[10])*(1-x[11]))*x[12]
  return(y)
}

phi_prime1<-function(x){
  y=x[5]
  return(y)
}

phi_prime2<-function(x){
  y=1-((1-x[8])*(1-x[9]))
```

```
  return(y)
}

  ##La fonction de structure du système
phi<-function(x){
  y= (1-phi_prime1(x))*(1-phi_prime2(x))*phi1(x)+phi_prime1(x)*(1-phi_prime2(x))*phi2(x)
  +(1-phi_prime1(x))*phi_prime2(x)*phi3(x)+phi_prime1(x)*phi_prime2(x)*phi4(x)
  return(y)
}

## Question 3
fc_IFRA <- function(lambda, beta, T, ln=F, test=F){
  vec <- seq(0, T, 0.01)
  k=1
  for(t in seq(0, T, 0.01)){
    vec[k] <- fc_survie(t, lambda, beta, test)
    if(ln){
      vec[k] <- -log(vec[k])/t
    }
    k=k+1
  }
  vec
}

z<-fc_IFRA(1.5,1.5,10,ln=T)
z1<-fc_IFRA(1.5,1.2,10,ln=T)
z2<-fc_IFRA(1.5,5,2.5,ln=T)
z3<-fc_IFRA(1.5,1.9,10,ln=T)
par(mfrow = c(2,2), mar = c(1,1,1,1))
plot(z,type="l", xlab = "durée t", ylab = "y",col="red")
text(800,3,"beta=1.5",font=3)
plot(z1,type="l", xlab = "durée t", ylab = "y",col="red")
text(800,3,"beta=1.2",font=3)
plot(z2,type="l", xlab = "durée t", ylab = "y",col="red")
text(400,3,"beta=2.5",font=3)
plot(z3,type="l", xlab = "durée t", ylab = "y",col="red")
text(400,4,"beta=1.9",font=3)

## Question 4

VectCrea <- function(w, lambda, beta, alpha){
  t <- seq(0, w, 0.01)
  n <- length(t)
  R <- rep(0, n)
  X <- rep(1,n)
  u <- rep(0,n)
  for(i in 1:n){
    y <- exp(-(t[i]/lambda)^beta)
    R[i] <- y
    if(i>1){
      u[i] <- runif(1,0,1)
      if(X[i]==2){
        X[i:n] <- VectCrea(ceiling((n-i)/100), lambda, beta, alpha)
```

```
        break
      }
      else{
        if(X[i-1]==1){
          if(u[i]>R[i]){X[i] <- 0}
          else{ X[i] <- 1 }
        }
        if(X[i-1]==0){
          h <- ceiling(rexp(1, alpha))
          X[i] <- 0
          if(i+h<=n){
            X[i+h] <- 2
          }
        }
      }
    }
  }
  return(X)
}

Traject<-function(w,lambda,beta,alpha,est=T){
  X <- matrix(rep(0,w*12), w, 12)
  for(i in 1:12){
    X[,i] <- VectCrea(w, lambda, beta, alpha)
  }
  Xt <- rep(1,n)
  for(i in 1:n){
    Xt[i] <- phi(X[i,])
  }

  if(est){
    sum(Xt)/100
  }
  else{
    plot(seq(0, w, 0.01), Xt, type="l", col="red")
  }
}

par(mfrow = c(2,2), mar = c(1,1,1,1))
Traject(10,10,3,0.01, est=F)
text(8,0.9,"beta=3",font=2)

Traject(10,10,4,0.01,est=F)
text(8,0.9,"beta=4",font=2)

Traject(10,10,6,0.01,est=F)
text(8,0.9,"beta=6",font=2)

Traject(10,10,8,0.01,est=F)
text(8,0.9,"beta=8",font=2)

## Question 5
```

```r
NbPanne <- function(w,lambda,beta,alpha){
  Nb <- 0
  t <- seq(0, w, 0.01)
  n <- length(t)
  X <- matrix(rep(0,n*12), n, 12)
  for(i in 1:12){
    X[,i] <- VectCrea(w, lambda, beta, alpha)
  }
  Xt <- rep(1,n)
  for(i in 1:n){
    Xt[i] <- phi(X[i,])
    if(i>1){
      if(Xt[i]==0 & Xt[i-1]==1){
        Nb <- Nb+1
      }
    }
  }
  return(Nb)
}

NbPanne(10,5,3,0.01)

TpsFonction <- function(w,lambda,beta,alpha){
  Y <- 0
  t <- seq(0, w, 0.01)
  n <- length(t)
  X <- matrix(rep(0,n*12), n, 12)
  for(i in 1:12){
    X[,i] <- VectCrea(w, lambda, beta, alpha)
  }
  Xt <- rep(1,n)
  for(i in 1:n){
    Xt[i] <- phi(X[i,])
    if(Xt[i]==1){
        Y <- Y+1
    }
  }
  return(Y/100)
}

TpsFonction(10,5,3,0.01)

### Lois des grand nombres et TCL

A <- rep(1,100)
for(i in 1:100){
  A[i] <- NbPanne(i,10,2,1)/i
}
plot(A, type="l")

A <- rep(1,100)
for(i in 1:100){
  A[i] <- TpsFonction(i,10,2,1)/i
```

```
}
plot(A, type="l")

A <- rep(1,100)
for(i in 1:100){
  A[i] <- (NbPanne(i,10,2,1)-1.5*i)/sqrt(i)
}
hist(A)

## Question 6

VectCrea2 <- function(w, lambda, beta, TpsAttente){
  t <- seq(0, w, 0.01)
  n <- length(t)
  R <- rep(0, n)
  X <- rep(1,n)
  u <- rep(0,n)
  for(i in 1:n){
    y <- exp(-(t[i]/lambda)^beta)
    R[i] <- y
    if(i>1){
      u[i] <- runif(1,0,1)
      if(X[i]==2){
        X[i:n] <- VectCrea2(ceiling((n-i)/100), lambda, beta, TpsAttente)
        break
      }
      else{
        if(X[i-1]==1){
          if(u[i]>R[i]){X[i] <- 0}
          else{ X[i] <- 1 }
        }
        if(X[i-1]==0){
          X[i] <- 0
          if(i+TpsAttente<=n){
            X[i+TpsAttente] <- 2
          }
        }
      }
    }
  }
  return(X)
}

Traject2<-function(w, lambda, beta, TpsAttente, est=T){
  t <- seq(0, w, 0.01)
  n <- length(t)
  X <- matrix(rep(0,n*12), n, 12)
  for(i in 1:12){
    X[,i] <- VectCrea2(w, lambda, beta, TpsAttente)
  }
  Xt <- rep(1,n)
  for(i in 1:n){
    Xt[i] <- phi(X[i,])
```

```
  }

  if(est){
    sum(Xt)/100
  }
  else{
    plot(seq(0, w, 0.01), Xt, type="l", col="red")
  }
}

Traject2(10,10,4,100, est=F)
```