# Projet-RM04

Baptiste Toussaint

2024-04-12

## Contents

## Exercice 2

```r
# Pour régler la génération aléatoire et conserver les mêmes
# valeurs à la compilation
set.seed(54684)

# Block de code pour effacer le contenu de la mémoire, utile à la compilation
rm(list = ls())
```

On a un système $S$ à $n$ composants $iid$ de loi exponentielle. On a $\lambda = 0.1(jour^{-1})$ en configuration parallèle.

On peut procéder ainsi pour simuler un instant de panne de $S$ unique pour $n$ fixé :

1. On simule n réalisation $T_1, ..., T_n$ aléatoires de la loi exponentielle de paramètre lambda
2. On calcule $T_S = max(T_1, ..., T_n)$ (car la structure est parallèle).

```r
# Ce bloc de code permet d'écrire des fonctions utiles pour la suite le
# l'exercice.

# Le 'pipe' %>% permet de passer des arguments à des fonctions en R.
# On peut par exemple écrire :
#   10 %>% exp()
# Pour calculer exp(10).

exe2_simu_panne_systeme <- function(n, taux) {

    # Retourne une réalisation de panne du système pour n
    # et pour lambda = taux

    # crée n tirage de loi exponentiel avec lambda = taux, calcule le maximum
    # et retourne le résultat.
    n %>% rexp(taux) %>% max() %>% return()
}

exe2_simu_mttf_system <- function(n,taux,p) {

    # Calcul un MTTF théorique à partir de p réplications pour n composants
    # et un taux.

    # Crée p réplications de date de panne pour le système.
    # Calcule la moyenne de ces p réplications et retourne le résultat.
    replicate(p,exe2_simu_panne_systeme(n,taux)) %>% mean() %>% return()
}

# Calcul théorique du MTTF avec le résultat connu pour ce système
# spécifique.
exe2_theoric_mttf_system <- function(n, taux) {return(sum(1/1:n)/taux)}
```

La fonction `exe2_simu_panne_systeme` ci-dessus retourne une réalisation de panne du système pour $n$ et $\lambda$.

On peut essayer pour les paramètres suivants : $n = 10$ et $\lambda = 0.1$.

```r
exe2_simu_panne_systeme(10,0.1)
```

```
## [1] 32.16055
```

Dans ce cas, le système tombe donc en panne pour 32,16 jours.

Pour rappel, pour un tel système, on a le résultat théorique suivant :

$$MTTF_n = \frac{1}{\lambda} \sum_{i=1}^{n} \frac{1}{i}$$

Si on fait l'application numérique avec $n = 10$ et $\lambda = 0.1$ :

```
exe2_theoric_mttf_system(10,0.1)
```

```
## [1] 29.28968
```

On trouve donc : $MTTF_{n=10}(\lambda = 0.1) \simeq 29.29$.

Ensuite on va chercher à évaluer l'évolution du $MTTF$ en fonction de $n$ et comparer les simulations avec les résultats théoriques.

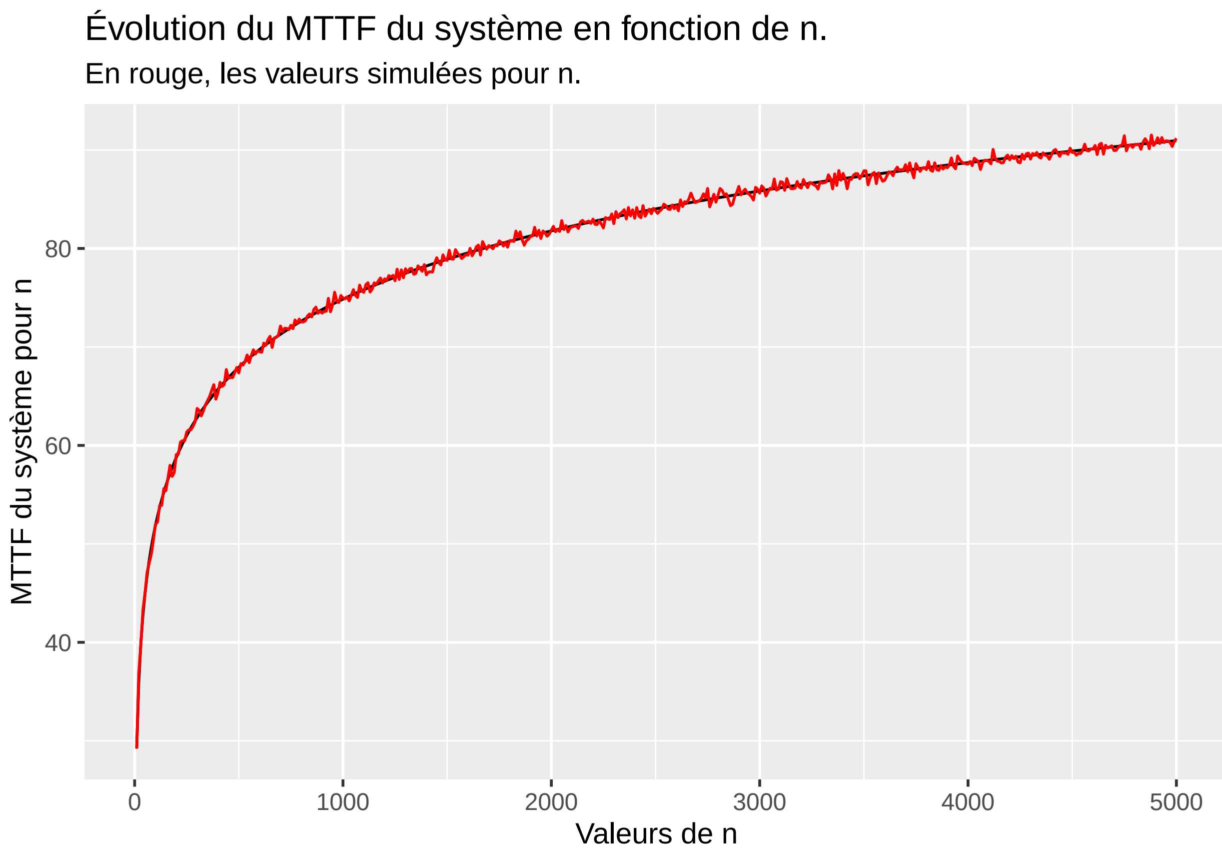
Pour trouver le $MTTF$ par simulation, il suffit de répliquer la fonction : `exe2_simu_panne_systeme` $p$ fois (avec $p$ relativement grand) et de prendre la moyenne des simulations réalisées, ce que fait la fonction `exe2_simu_mttf_system`.

On va donc appeler cette fonction pour plusieurs valeurs de $n$ pour constituer une liste de $MTTF$ simulés en fonction de $n$.

```
p = 1000 # nombre de répliques par simulations pour le calcul du MTTF
taux = 0.1
nvals = seq(10,5000,10)

# L'object contient les simulations de MTTF pour n de 10 à 5000
# par pas de 10
# Un peu long à la compilation

exe2_simu_MTTF_n <- nvals %>% mapply(FUN = exe2_simu_mttf_system, taux = taux, p = p)

# L'object contient les valeurs théoriques de MTTF pour n de 10 à 5000
# par pas de 10
# Un peu long à la compilation
exe2_theoric_MTTF_n <- mapply(FUN = exe2_theoric_mttf_system,
                              n = nvals, taux = 0.1)
```

On peut tracer les courbes :

```
ggplot() +
    geom_line(aes(x = nvals, y = exe2_theoric_MTTF_n)) +
    geom_line(aes(x = nvals, y = exe2_simu_MTTF_n),
              color = "red") +
    labs(x = "Valeurs de n",
         y = "MTTF du système pour n",
         title = "Évolution du MTTF du système en fonction de n.",
         subtitle = "En rouge, les valeurs simulées pour n.")
```



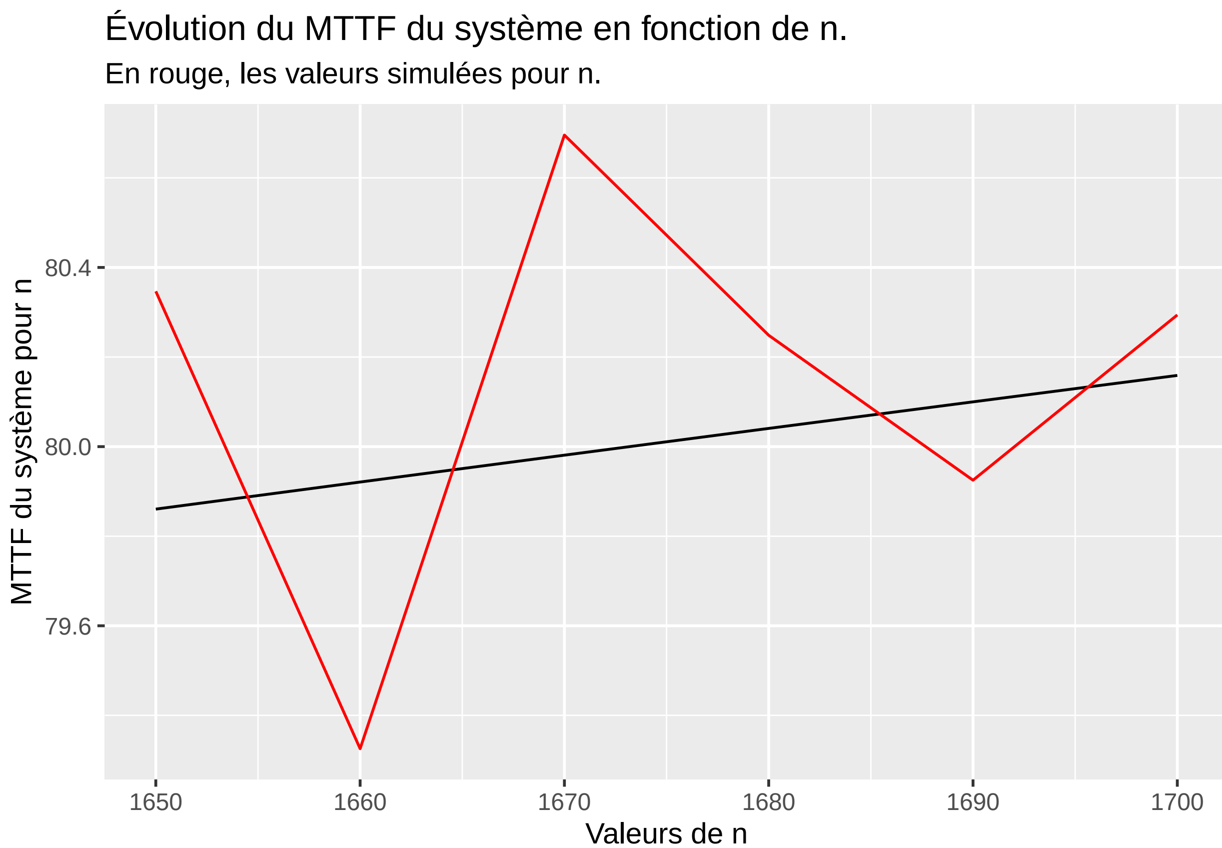

On observe donc que les valeurs simulés sont bel et bien proches des valeurs théoriques.

On remarque que le $MTTF$ croit rapidement mais il semble ensuite prendre une forme logarithmique et donc une croissance très très lente.

Graphiquement, il semble qu'l'on attend un $MTTF$ de 80 jours pour une valeur de $n$ proche de $n = 1700$.

On peut essayer de zoomer sur le grahe pour s'en assurer :

```
zoom = 165:170
ggplot() +
    geom_line(aes(x = nvals[zoom], y = exe2_theoric_MTTF_n[zoom])) +
    geom_line(aes(x = nvals[zoom], y = exe2_simu_MTTF_n[zoom]),
              color = "red") +
    labs(x = "Valeurs de n",
         y = "MTTF du système pour n",
         title = "Évolution du MTTF du système en fonction de n.",
         subtitle = "En rouge, les valeurs simulées pour n.")
```




On remarque qu'on atteint très précisément un $MTTF$ de 80 pour un $n$ entre 1670 et 1675.

Celà signifie qu'il faudrait environ 1670 composants en série pour s'assurer que le système tombe en panne dans environ 80 jours.

Ce résultat peut être approché par le calcul. En effet on utilise l'approximation :

$$\sum_{i=1}^{n} \frac{1}{i} \simeq log(n) + 0.577$$

Dans ce cas :

$$MTTF_n \simeq \frac{1}{\lambda}(log(n) + 0.577)$$

On peut donc résoudre l'équation pour $n^*$:

$MTTF_n \simeq \frac{1}{\lambda}(log(n) + 0.577) \simeq 80$

$log(n) \simeq 80\lambda - 0.577$

$n \simeq e^{80\lambda - 0.577}$

```
exp(80*0.1-0.577)
```

```
## [1] 1674.048
```

On retrouve une très bonne approximation qui nous indique que pour avoir un système qui tombe en panne au bout de 80 jours il nous faut environ 1674 composants.

# Exercice 3

## Partie 1

Dans cette première partie nous allons étudier 6 systèmes différents :

- le système élémentaire simple A/B ;
- le système avec B en redondance uniquement ;
- le système en redondance haut-niveau ;
- le système en redondance bas niveau ;
- le système en redondance passive haut-niveau ;
- le système en pont.

Pour chacun de ces systèmes nous allons :

- calculer l'expression de $R_{sys}$ et $MTTF_{sys}$ ;
- effectuer l'application numérique pour trouver la valeur du $MTTF_{sys}$ ;
- simuler une panne du système ;
- simuler $p$ panne du système pour calculer une approximation du $MTTF_{sys}$.

```
panne_comp_exp <- function(lambda = 0.1) {

    # Simule la date de panne d'un composant de loi exponentielle
    # et de paramètre lambda. date en annnée
    # Revient à renommer la fonctions le base de R pour plus de lisibilité dans
    # le rapport

    return(rexp(1,lambda))
}
```

#### Système élémentaire simple A/B

```
# Données
lambda_A = 1 # ans-1
lambda_B = 5 # ans-1

# nombre de réplications
p = 10000

message_mttf_simule_a = "MTTF simulé en années : %f"
message_mttf_simule_j = "MTTF simulé en jours : %f"
```

Dans un premier temps a un système simple avec deux composants $A$ et $B$ en série.

Ce cas est simple puisqu'on a :

$$MTTF_{sys} = \frac{1}{\sum_{i \in C} \lambda_i}$$

On a :

- $T_{sys_1} = min(T_A, T_B)$ ;
- $R_{sys_1} = e^{-(\lambda_A + \lambda_B)t}$ ;
- $MTTF_{sys_1} = \frac{1}{\lambda_A + \lambda_B}$ ;
- $[A.N] MTTF_{sys_1} = \frac{1}{6} ans^{-1} \simeq 60.83 j^{-1}$

On peut simuler ce système :

```
exe3_sys1_panne <- function(rate_A, rate_B) {

    # Retourne la date de panne d'un système en série de deux composants A
    # Et B de loi exponentielle.

    return(min(panne_comp_exp(rate_A),panne_comp_exp(rate_B)))
}
```

Dans ce cas on peut maintenant estimer le $MTTF$ de ce système par simulation :

```
simulation <- replicate(p,exe3_sys1_panne(lambda_A,lambda_B)) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 0.169248"
```

```
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 61.775435"
```

#### Système avec B en redondance uniquement

On a donc un système série/parallèle. Dans ce cas :

- $T_{sys_2} = min(T_A, max(T_B, T'_B))$
- $R_{sys_2} = e^{-\lambda_A t}(2e^{-\lambda_B t} - e^{-2\lambda_B t}) = 2e^{-(\lambda_A+\lambda_B)t} - e^{-(2\lambda_B+\lambda_A)t}$
- $MTTF_{sys_2} = \frac{2}{\lambda_A+\lambda_B} - \frac{1}{2\lambda_B+\lambda_A}$
- $[A.N]MTTF_{sys_2} \simeq 88.48j$

Si l'on fait l'application numérique on a en théorie :

```
# MTTF théorique en jours
((2/(5+1))-(1/(2*5+1)))*365
```

```
## [1] 88.48485
```

Le système tombe en moyenne en panne après 88.48 jours.

Par la simulation :

```
exe3_sys2_panne <- function(rate_A,rate_B) {
    min(panne_comp_exp(rate_A),
        max(panne_comp_exp(rate_B),panne_comp_exp(rate_B))) %>%
        return()
}
```

```
simulation <- replicate(p,exe3_sys2_panne(lambda_A,lambda_B)) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 0.244062"
```

```
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 89.082753"
```

On retrouve bien un résultat proche de la valeur théorique.

### Système en redondance haut-niveau

Dans ce cas on a un système parallèle/série avec :

- $T_{sys_3} = max(min(T_A, T_B), min(T'_A, T'_B))$
- $R_{sys_3} = 2e^{-(\lambda_A+\lambda_B)t} - e^{-2(\lambda_A+\lambda_B)t}$
- $MTTF_{sys_3} = \frac{2}{\lambda_A+\lambda_B} - \frac{1}{2(\lambda_A+\lambda_B)}$
- $[A.N]MTTF_{sys_3} = \frac{2}{3} - \frac{1}{12} = 0.25ans^{-1} = 91.25j^{-1}$

Dans ce cas on a un $MTTF$ théorique de :

```
(2/(lambda_A+lambda_B))-(1/(2*(lambda_A+lambda_B))) # En ans
```

```
## [1] 0.25
```

```
((2/(lambda_A+lambda_B))-(1/(2*(lambda_A+lambda_B))))*365 # En jours
```

```
## [1] 91.25
```

Par la simulation on trouve :

```
exe3_sys3_panne <- function(rate_A,rate_B) {
    max(
        min(panne_comp_exp(rate_A),panne_comp_exp(rate_B)),
        min(panne_comp_exp(rate_A),panne_comp_exp(rate_B))
    ) %>%
        return()
}
```

```
simulation <- replicate(p,exe3_sys3_panne(lambda_A,lambda_B)) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 0.250382"
```

```
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 91.389605"
```

### Système en redondance bas niveau

Dans ce cas on a un système parallèle/série avec :

- $T_{sys_4} = min(max(T_A, T'_A), max(T_B, T'_B))$
- $$\begin{aligned} R_{sys_4}(t) &= (1-(1-R_A(t))^2)(1-(1-R_B(t))^2) \\ &= (R_A(t)^2 - 2R_A(t))(R_B(t)^2 - 2R_B(t)) \\ &= e^{-2(\lambda_A+\lambda_B)t} - 2e^{-(2\lambda_A+\lambda_B)t} - 2e^{-(\lambda_A+2\lambda_B)t} + 4e^{-(\lambda_A+\lambda_B)t} \end{aligned}$$
- $$MTTF_{sys_4} = \frac{4}{\lambda_A+\lambda_B} - \frac{2}{\lambda_A+2\lambda_B} - \frac{2}{2\lambda_A+\lambda_B} + \frac{1}{2(\lambda_A+\lambda_B)}$$
- $$\begin{aligned} [A.N]MTTF_{sys_4} &= \frac{4}{6} - \frac{2}{11} - \frac{2}{7} + \frac{1}{12} \\ &\simeq 0.2824ans \\ &\simeq 103.10j \end{aligned}$$

```r
(4/(lambda_A+lambda_B))-(2/(lambda_A+2*lambda_B))-(2/(2*lambda_A+lambda_B))+
    (1/(2*(lambda_A+lambda_B))) # En ans
```

```
## [1] 0.2824675
```

```r
((4/(lambda_A+lambda_B))-(2/(lambda_A+2*lambda_B))-(2/(2*lambda_A+lambda_B))+
    (1/(2*(lambda_A+lambda_B)))
)*365 # En jours
```

```
## [1] 103.1006
```

Par la simulation on trouve :

```r
exe3_sys4_panne <- function(rate_A,rate_B) {
    min(
        max(panne_comp_exp(rate_A),panne_comp_exp(rate_A)),
        max(panne_comp_exp(rate_B),panne_comp_exp(rate_B))
    ) %>%
        return()
}
```

```r
simulation <- replicate(p,exe3_sys4_panne(lambda_A,lambda_B)) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 0.280923"
```

```r
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 102.536971"
```

#### Système en redondance passive haut-niveau

Ce système peut être assimilé à deux système élémentaires A/B (système n°1) mit en redondance passive.

Dans ce cas on a :

- $T_{sys_5} = min(T_A, T_B) + min(T'_A, T'_B)$

Le calcul de $R_{sys_5}$ est plus complexe puisqu'il faut conditionner selon la valeur de panne du premier système.

On pose : $T_{sys_5} = X + Y$ avec $X = min(T_A, T_B)$ et Y = min(T_A',T_B')$. Attention, dans ce cas $X$ et $Y$ ont la même loi, identique à celle du système n°1 vu plus haut. Donc : $R_X = R_Y = R_{sys_1} = e^{-(\lambda_A+\lambda_B)t}$.

On a alors :

$$\begin{aligned}
P(T_{sys_5} \geq t) &= P(X + Y \geq t) \\
&= P(X \geq t)P(Y \geq t - X|X \geq t) + P(X \leq t)P(Y \leq t - X|X \leq t) \\
&Ona : P(Y \geq t - X|X \geq t) car X \geq t \Rightarrow t - X \leq 0 et P(Y \geq 0) = 1 car \Omega_Y = \mathbb{R}^+ \\
&= R_X(t) + \int_{u \in \Omega_X, u \leq t} R_Y(t-u) f_X(u) du \\
&= e^{-(\lambda_A+\lambda_B)t} + \int_0^t e^{-(\lambda_A+\lambda_B)(t-u)} (\lambda_A + \lambda_B) e^{-(\lambda_A+\lambda_B)u} du \\
&= e^{-(\lambda_A+\lambda_B)t} + \int_0^t (\lambda_A + \lambda_B) e^{-(\lambda_A+\lambda_B)u} du \\
&= e^{-(\lambda_A+\lambda_B)} + (\lambda_A + \lambda_B) e^{-(\lambda_A+\lambda_B)t} t \\
&= e^{-(\lambda_A+\lambda_B)} (1 + (\lambda_A + \lambda_B) t)
\end{aligned}$$

On peut calculer le $MTTF_{sys_5}$ (On pose $\psi = \lambda_A + \lambda_B$):

$$
\begin{aligned}
MTTF_{sys_5} &= \int_0^{+\infty} e^{-\psi t}(1+\psi t)dt \\
&= \left[(1+\psi t)(-\frac{1}{\psi}e^{-\psi t})\right]_0^{\infty} - \int_0^{\infty} \psi(-\frac{1}{\psi}e^{-\psi t})dt \\
&= -\frac{1}{\psi}\left[e^{-\psi t} + \psi e^{-\psi t}t\right]_0^{\infty} + \int_0^{\infty} e^{-\psi t}dt \\
&= \frac{1}{\psi} - \left[-\frac{1}{\psi}e^{-\psi t}\right]_0^{\infty} \\
&= \frac{1}{\psi} - \frac{1}{\psi}(0-1) \\
&= \frac{2}{\psi} = \frac{2}{\lambda_1 + \lambda_B} = 2MTTF_{sys_1}.
\end{aligned}
$$

On a alors :

$$[A.N] MTTF_{sys_5} = \quad = \frac{2}{6} = \frac{1}{3} ans \simeq 121.67j$$

Par la simulation on trouve :

```r
exe3_sys5_panne <- function(rate_A,rate_B) {
    min(panne_comp_exp(rate_A),panne_comp_exp(rate_B)) +
        min(panne_comp_exp(rate_A),panne_comp_exp(rate_B)) %>%
        return()
}
```

```r
simulation <- replicate(p,exe3_sys5_panne(lambda_A,lambda_B)) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 0.336252"
```

```r
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 122.731821"
```

**Système en pont avec $\lambda_C = \lambda_B$**

Dans ce système en pont, on trouve au centre un composant $C$ de loi identique à $B$ donc $\lambda_C = \lambda_B$.

Système pont

Dans ce cas plusieurs options s'offrent à nous.

Pour calculer la valeur de $R_{sys_6}$ il faut passer par un conditionnement selon la valeur du composant $C$

- Si $C$ est en marche alors on a $R_{sys_6}(t) = R_{sys_4}(t)$
- Si $C$ est panne alors on a $R_{sys_6}(t) = R_{sys_4}(t)$

On a alors :

$$
\begin{aligned}
R_{sys_6}(t) &= P(T_{sys_6} \geq t) = P(T_C \geq t)P(T_{sys_4} \geq t) + P(T_C \leq t)P(T_{sys_3} \geq t) \\
&= R_C(t)R_{sys_4}(t) + (1 - R_C(t))R_{sys_3}(t) \\
&= 2e^{-(\lambda_A+2\lambda_B)t} + 2e^{-(2\lambda_A+3\lambda_B)t} + 2e^{-(\lambda_A+\lambda_B)t} - 2e^{-(\lambda_A+3\lambda_B)t} - 3e^{-(\lambda_A+\lambda_B)t}
\end{aligned}
$$

Dans ce cas on a :

$$MTTF_{sys_6} = \frac{2}{\lambda_A + 2\lambda_B} + \frac{2}{2\lambda_A + 3\lambda_B} + \frac{2}{\lambda_A + \lambda_B} - \frac{2}{\lambda_A + 3\lambda_B} - \frac{3}{2(\lambda_A + \lambda_B)}$$

Soit :

$$[A.N] MTTF_{sys_6} = \frac{2}{11} + \frac{2}{17} + \frac{2}{6} - \frac{2}{16} - \frac{3}{12}$$
$$= \simeq 0.2578 ans \simeq 94.10 jours$$

On remarque alors que :

$$MTTF_{sys_5} > MTTF_{sys_4} > MTTF_{sys_6} > MTTF_{sys_3} > MTTF_{sys_2} > MTTF_{sys_1}$$

Si l'on utilise la simulation pour retrouver ce résultat on a alors :

$$T_{sys_6} = max(min(T_A, T_B), min(T_A, T'_B, T_C), min(T'_A, T'_B), min(T'_A, T_B, T_C))$$

```
exe3_sys6_panne <- function(rate_A,rate_B) {

A1 <- panne_comp_exp(rate_A)
A2 <- panne_comp_exp(rate_A)
B1 <- panne_comp_exp(rate_B)
B2 <- panne_comp_exp(rate_B)
C <- panne_comp_exp(rate_B)

  max(
      min(A1,B1),
      min(A1,B2,C),
      min(A2,B2),
      min(A2,B1,C)
  ) %>% return()

}
```

```
simulation <- replicate(p,exe3_sys6_panne(lambda_A,lambda_B)) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 0.261487"
```

```
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 95.442743"
```

## Partie 2

Dans le cas de ce dernier système en redondance bas niveau, les composants $A$ suivent une loi de Weibull $W(\theta = 100, \beta = 1.5)$ et les composants B suivent un lois normale : $N(\mu = 250, \sigma = 20)$.

Dans ce cas, en reprenant le résultat du système en redondance bas-niveau, on a :

$$R_{sys_7} = (2R_A(t) - R_A(t)^2)(2R_B(t) - R_B(t)^2)$$

En plus d'être pénible, le calcul de la fonction de survie $R_B$ fait intervenir la fonction d'erreur de Gauss $erf(x)$ car la fonction de répartition de la loi normale n'est pas clairement définie.

Il n'est donc pas possible d'obtenir une $MTTF$ théorique par le calcul et l'on passera alors par la simulation.

On a :

$$T_{sys_7} = min(max(T_A, T'_A), max(T_B, T'_B))$$

```
panne_comp_norm <- function(mu = 0, sigma = 1) {
    return(rnorm(1, mean = mu, sd = sigma))
}

panne_comp_weibull <- function(theta = 0.1, beta = 1) {
    return(rweibull(1, scale = theta, shape = beta))
}

exe3_sys6_panne <- function(param = list()) {

  min(
    max(panne_comp_weibull(param$theta,param$beta),
        panne_comp_weibull(param$theta,param$beta)),
    max(panne_comp_norm(param$mu,param$sigma),
        panne_comp_norm(param$mu,param$sigma))
  ) %>% return()

}
```

On peut alors simuler le $MTTF$ :

```
simulation <- replicate(
  p,exe3_sys6_panne(list(theta = 100, beta = 1.5, mu = 250, sigma = 20))) %>% mean()

sprintf(message_mttf_simule_a, simulation)
```

```
## [1] "MTTF simulé en années : 122.632615"
```

```
sprintf(message_mttf_simule_j, simulation * 365)
```

```
## [1] "MTTF simulé en jours : 44760.904599"
```

On a un $MTTF$ égal à 44760 jours.