TP 2 : Processus AR et MA, autocorrélogrammes

Paul Crespin

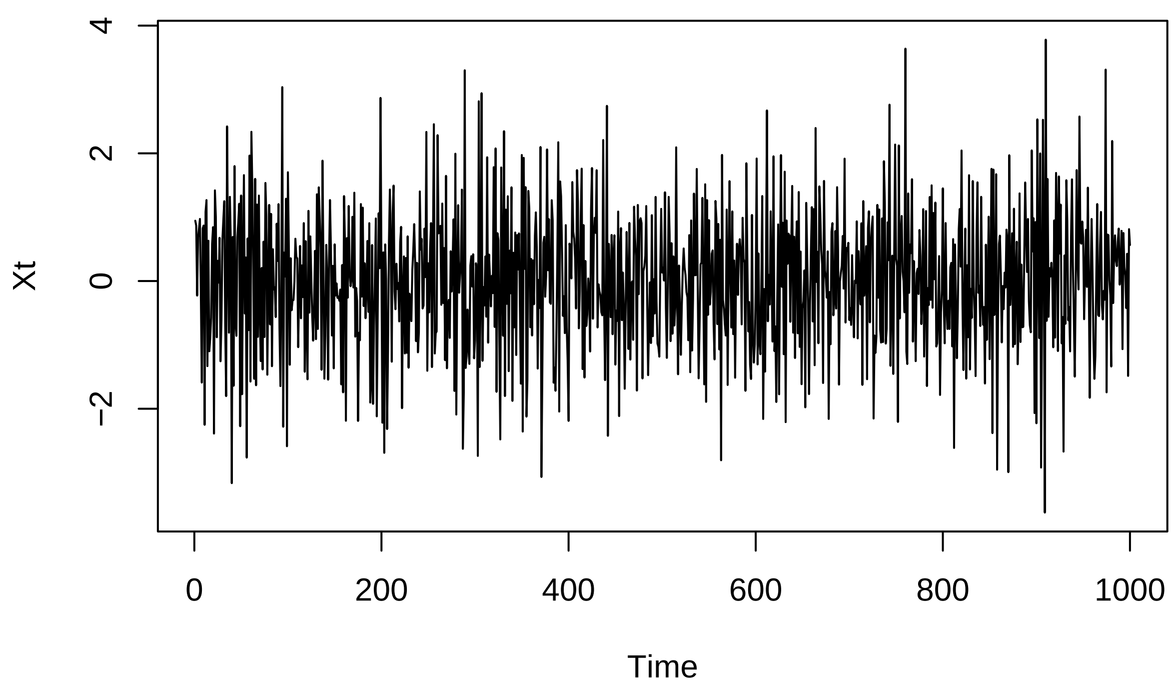
Les questions théoriques ne seront pas rédigées ici.

1. Simuler un processus

Simulation d'une processus MA

Question 1)

```
N=1000
BB <- rnorm(N+1)
theta=-1/3
Xt=BB[2:(N+1)]+theta*BB[1:N]
plot.ts(Xt)
```



On réalise ici un processus MA(1) défini selon la question 1.

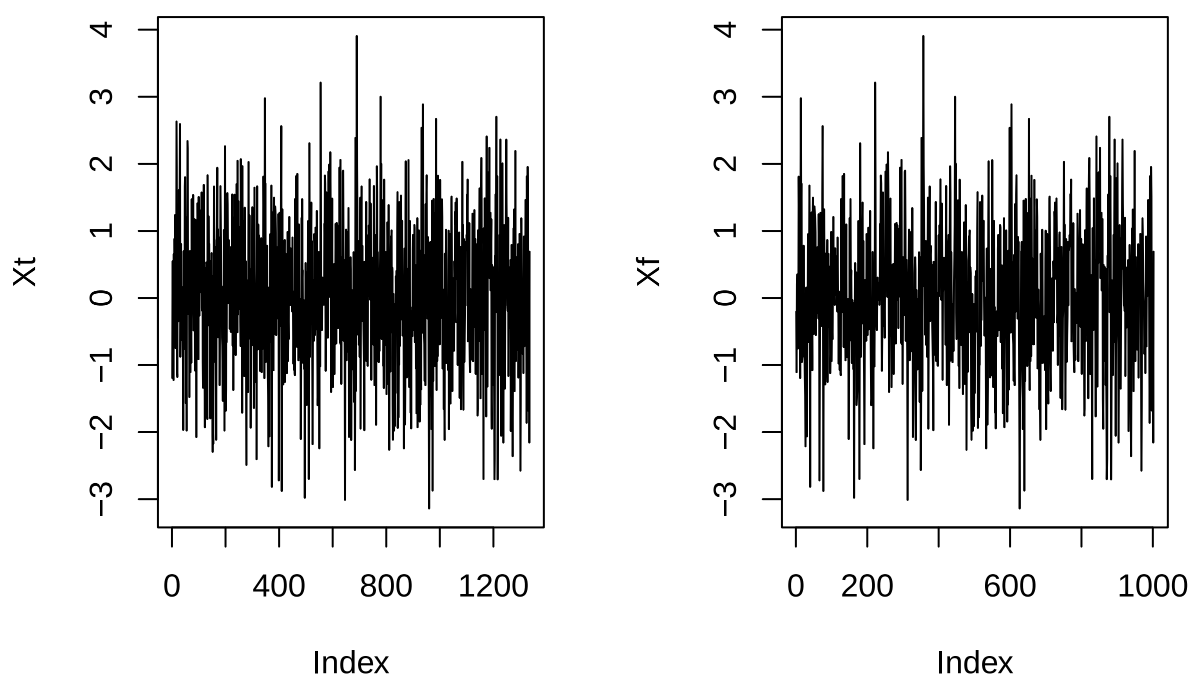
Simulation d'une processus AR

Question 2)

```
phi=c(4/15,-1/15)
init= runif(4)
NO= floor(N/3)
BB <- rnorm(N+NO+2)
Xt=phi[1]*init[2:4]+phi[2]*init[1:3]+BB[3:5]

for (i in 4:(N+NO+2)) {
  Xt[i] <- phi[1]*Xt[(i-1)]+phi[2]*Xt[(i-2)]+BB[i]
}

par(mfrow=c(1,2))
plot(Xt,type='l')
Xf=Xt[(NO+1):(N+NO+2)]
plot(Xf,type='l')
```

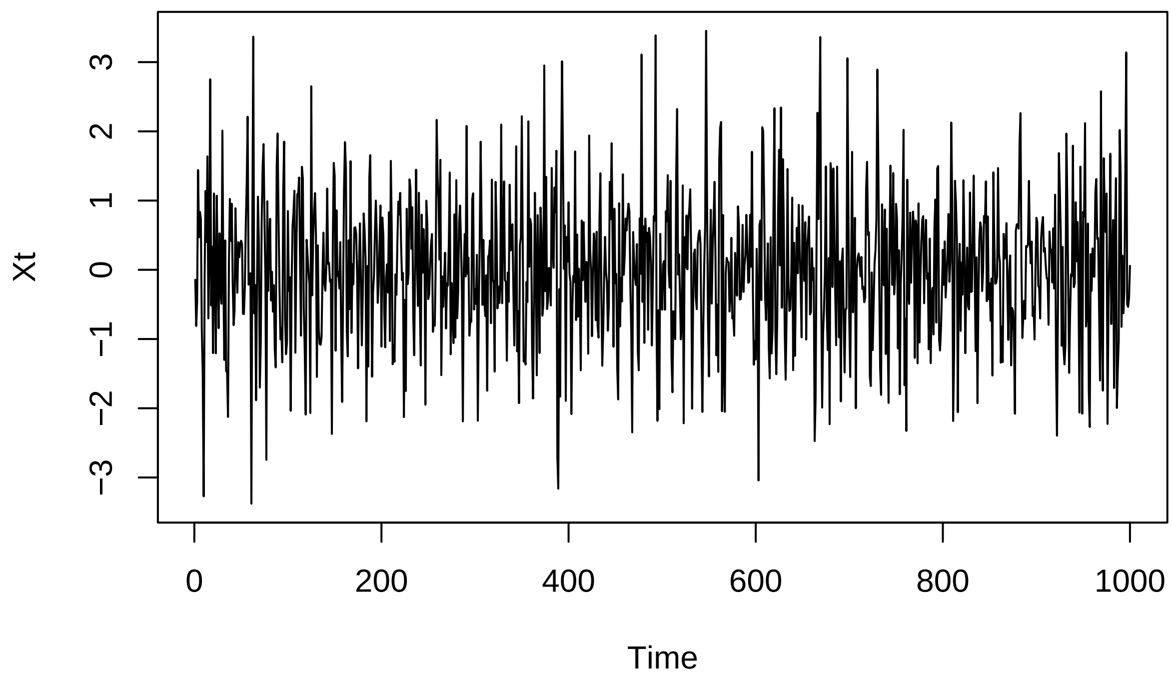


Ici, on réalise un processus AR(2) selon l'équation de la question 2. Le second graphique représente celui où on a éliminé les NO premières valeurs.

Simulation d'une processus ARMA avec arima.sim

Question 3:

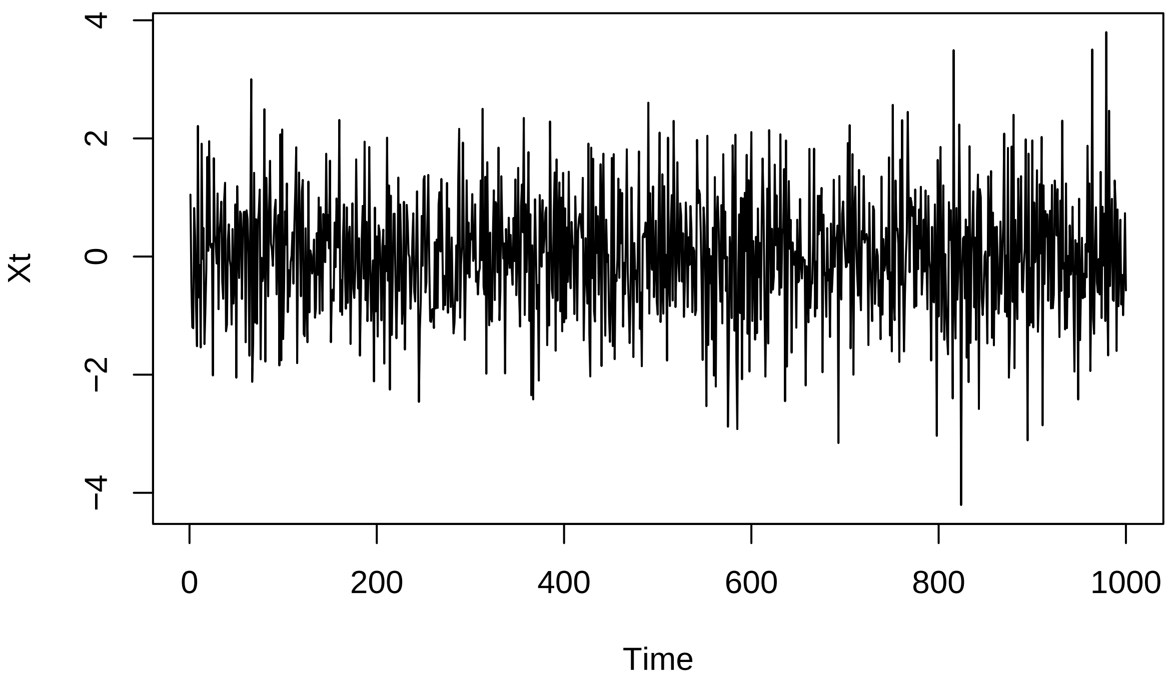
```
parametres = list(ar=c(1,-.25),ma=-1)
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
plot.ts(Xt)
```



On crée une liste des paramètres ar et ma; On crée notre série grâce à la fonction arima.sim avec les paramètres choisis, sur 1000 itérations avec les valeurs initiales qui sont faites avec une loi normale.

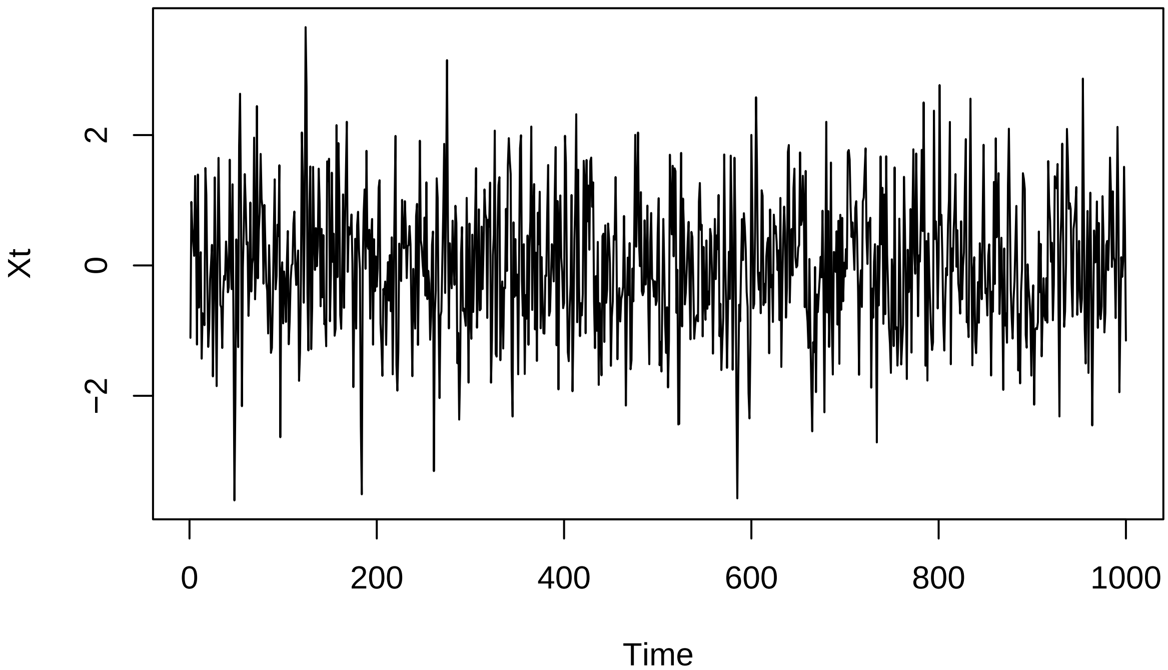
On va ensuite simuler des processus MA(1) et AR(2) avec les paramètres des équations 1 et 2, puis des AR(2), un MA(3) et un ARMA(2,2) avec des paramètres arbitraires.

```
parametres = list(ma=-1/3)
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
plot.ts(Xt)
```



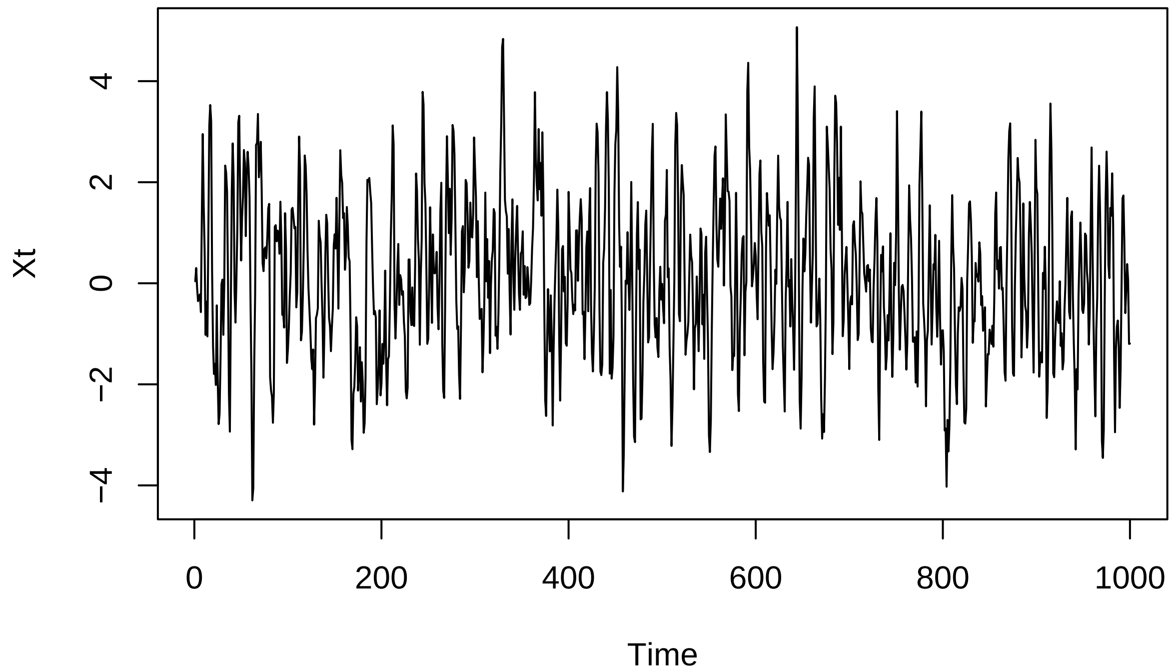
Un processus MA(1) avec paramètre -1/3.

```r
parametres = list(ar=c(4/15,-1/15))
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
plot.ts(Xt)
```



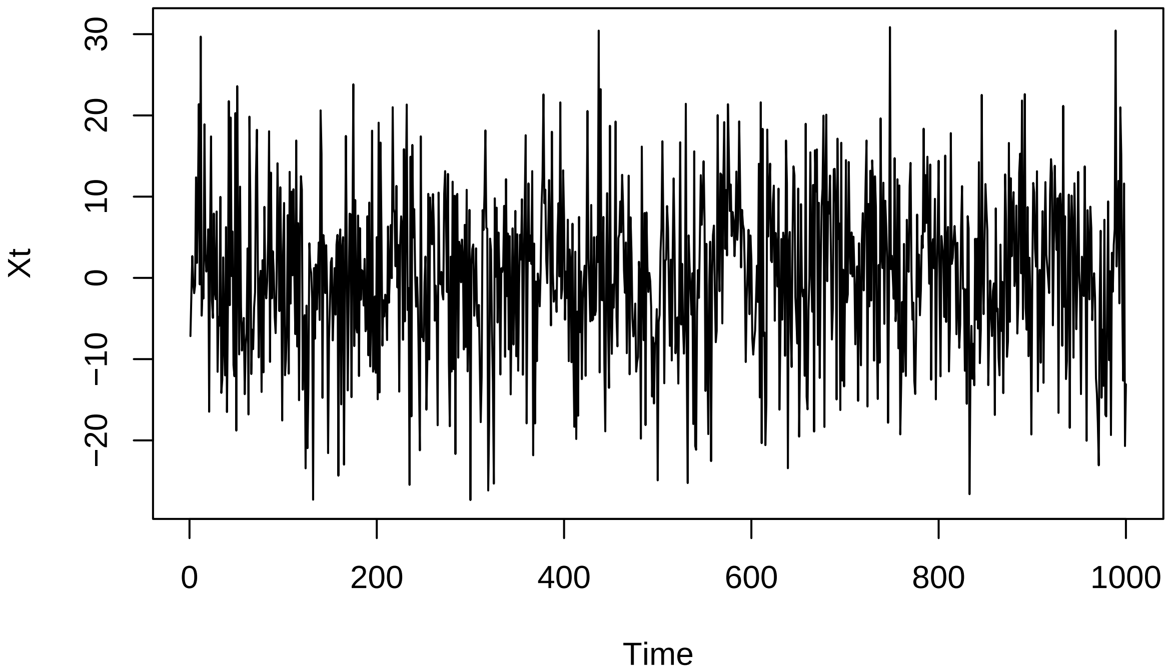
Un processus AR(1) avec paramètres 4/15 et -1/15.

```
parametres = list(ar=c(1,-.4))
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
plot.ts(Xt)
```



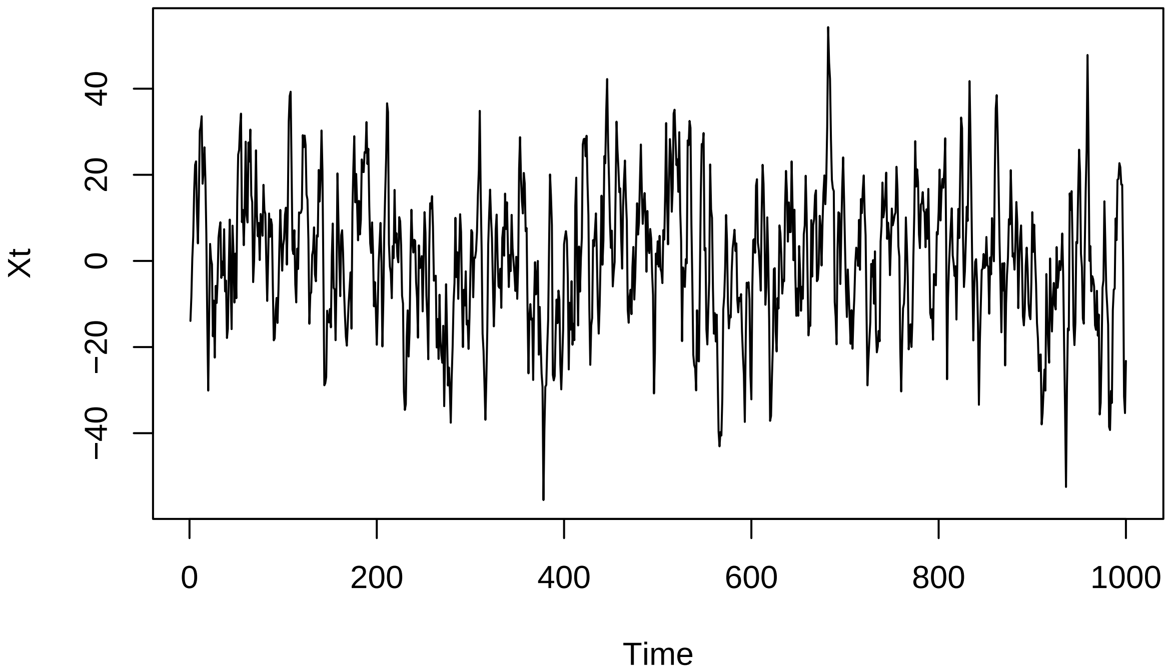
Un processus AR(2) avec paramètre 1 et -0.4.

```
parametres = list(ma=c(1,-.25,10))
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
plot.ts(Xt)
```



Un processus MA(3) avec paramètre 1, -0.25 et 10.

```r
parametres = list(ar=c(1,-.25),ma=c(-1,10))
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
plot.ts(Xt)
```



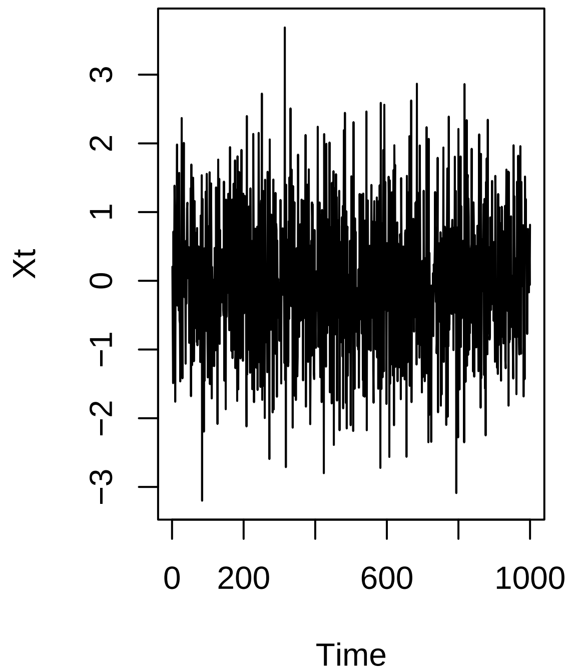
Un processus ARMA(2,2) avec paramètre 1, -1/4, -1 et 10.

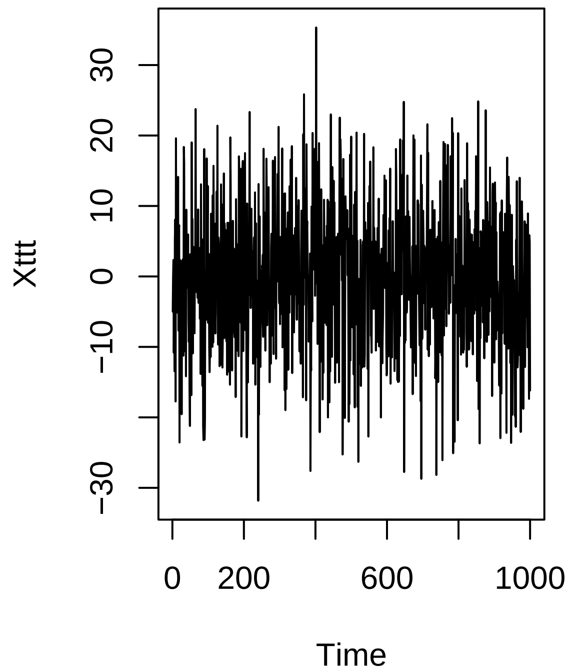
2) Fonctions d'autocorrélation et d'autocorrélation partielle

Fonction d'autocorrélation d'une processus MA

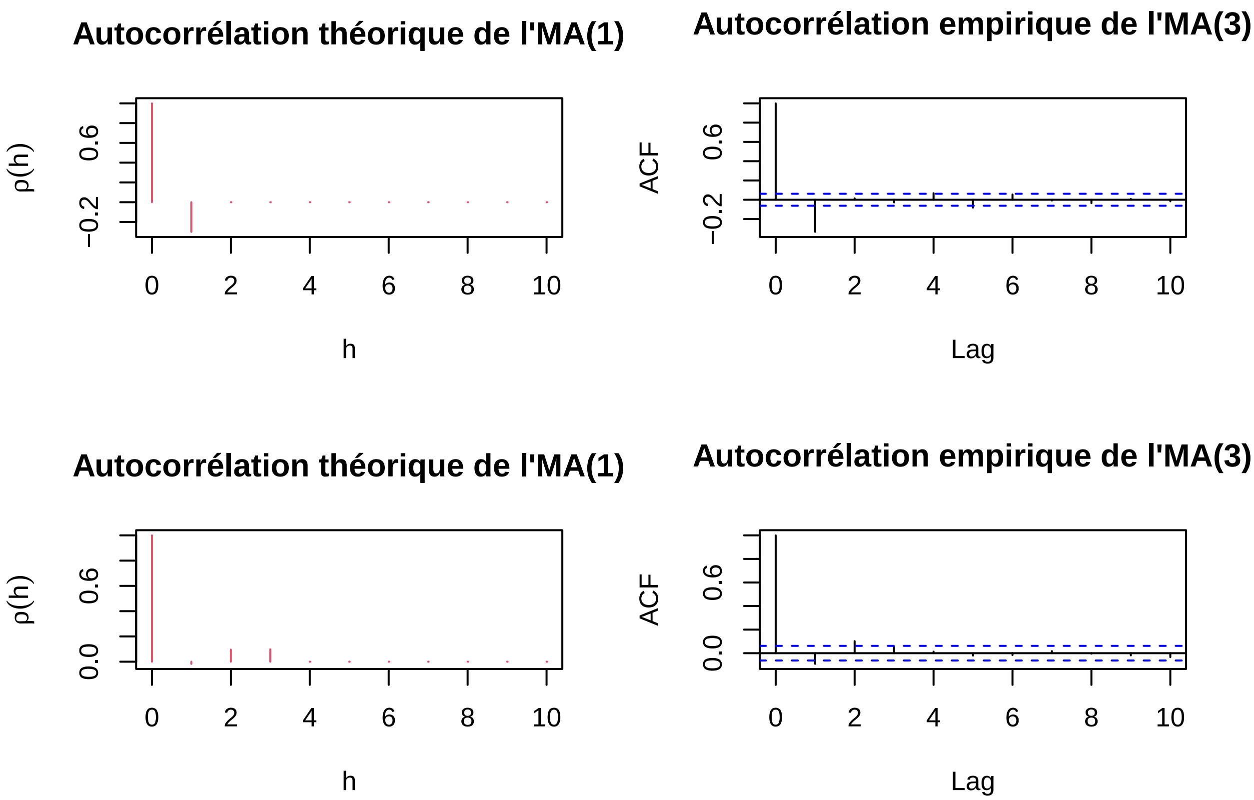
Question 4:

```
parametres = list(ma=-1/3)
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
parametres = list(ma=c(1,-.25,10))
Xttt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
par(mfrow=c(1,2))
plot.ts(Xt)
plot.ts(Xttt)
```





```
par(mfrow=c(2,2))
plot(0:10,c(1,-3/10,rep(0,9)),col=2,type='h',main="Autocorrélation théorique de l'MA(1)",xlab='h',ylab=
acf(Xt,type="correlation",main="Autocorrélation empirique de l'MA(3)",lag.max=10)
plot(0:10,c(1,(-1.75/(102+1/16)),(9.75/(102+1/16)),(10/(102+1/16)),rep(0,7)),col=2,type='h',main="Autoco
acf(Xttt,type="correlation",main="Autocorrélation empirique de l'MA(3)",lag.max=10)
```

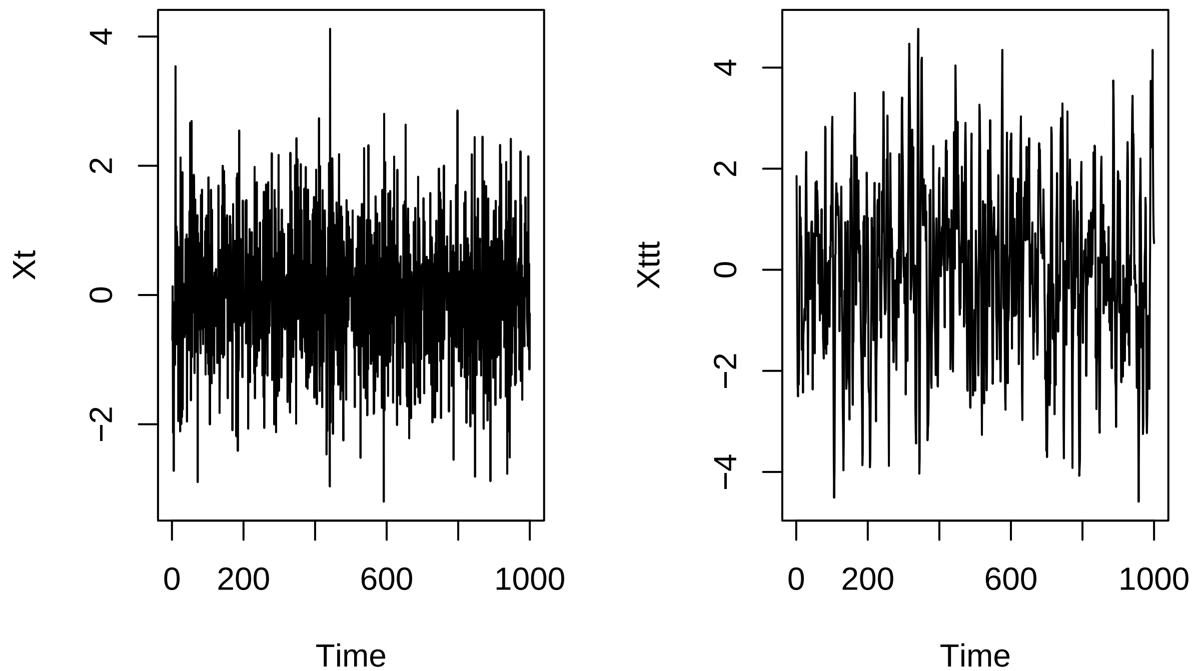


On identifie très facilement le théorique de l'empirique. Les rho sont calculés à la main selon la formule fournie. On regarde le dernier rho qui dépasse l'intervalle de confiance. Donc d'ordre 1 et 3.

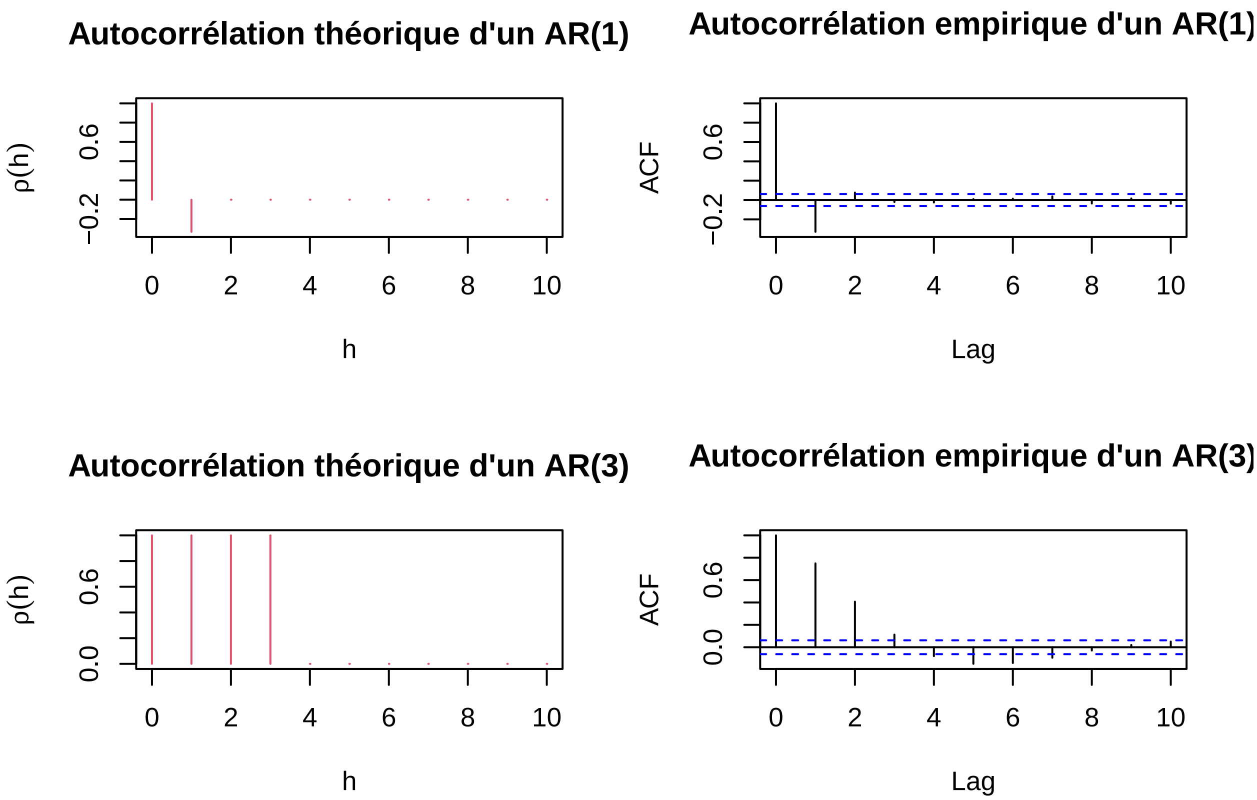
Fonction d'autocorrélation d'un processus AR

Question 6:

```
parametres = list(ar=-1/3)
Xt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
parametres = list(ar=c(1,-.25,-0.1))
Xttt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
par(mfrow=c(1,2))
plot.ts(Xt)
plot.ts(Xttt)
```



```
par(mfrow=c(2,2))
plot(0:10,c(1,-1/3,rep(0,9)),col=2,type='h',main="Autocorrélation théorique d'un AR(1)",xlab='h',ylab=e
acf(Xt,type="correlation",main="Autocorrélation empirique d'un AR(1)",lag.max=10)
plot(0:10,c(1,1,1,1,rep(0,7)),col=2,type='h',main="Autocorrélation théorique d'un AR(3)",xlab='h',ylab=
acf(Xttt,type="correlation",main="Autocorrélation empirique d'un AR(3)",lag.max=10)
```



On ne peut pas utiliser l'autocorrélation pour estimer l'ordre p d'un AR (contrairement à l'ordre q d'un AM), on voit la décroissante exponentielle pour l'AR(1) empirique et la sinusoïde amortie pour l'AR(3). Les deux graphiques d'autocorrélations sont de plus beaucoup plus éloignés.

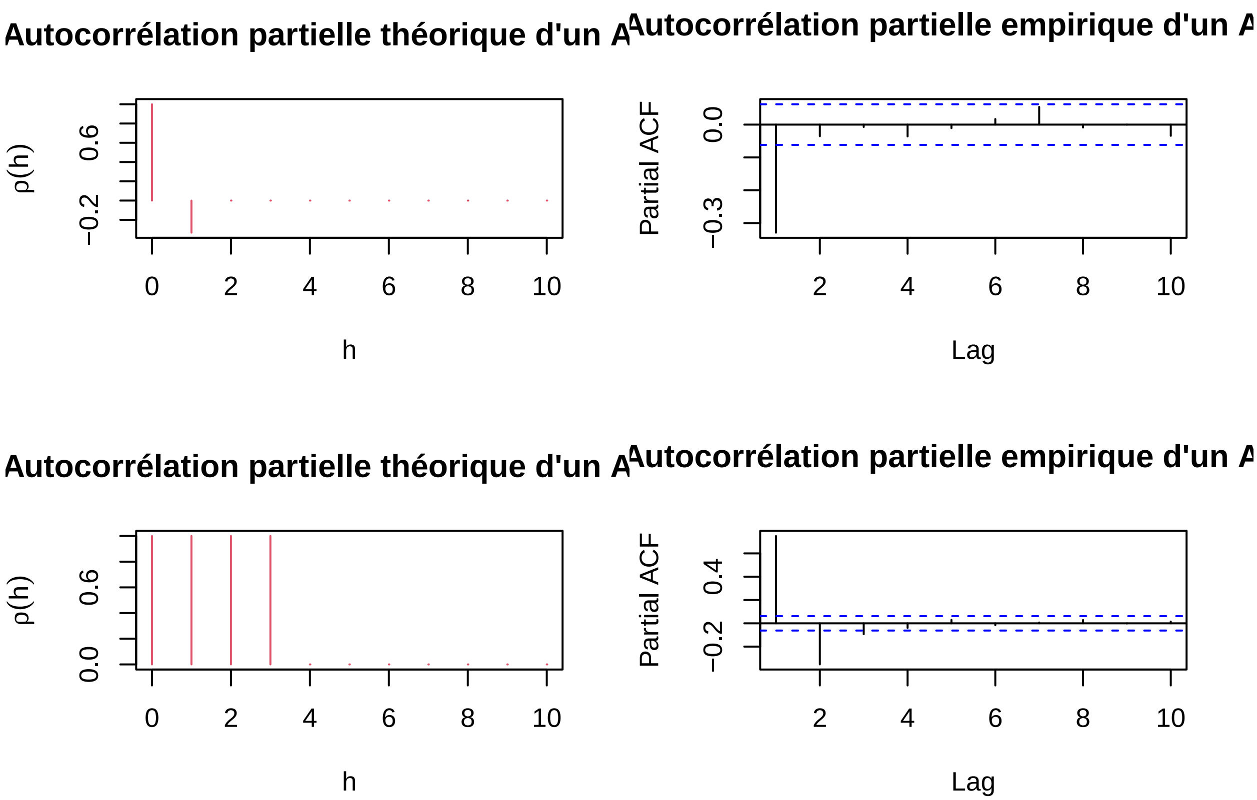
Fonction d’autocorrélation partielle

Fonction d’autocorrélation partielle d’une processus AR

Question 8:

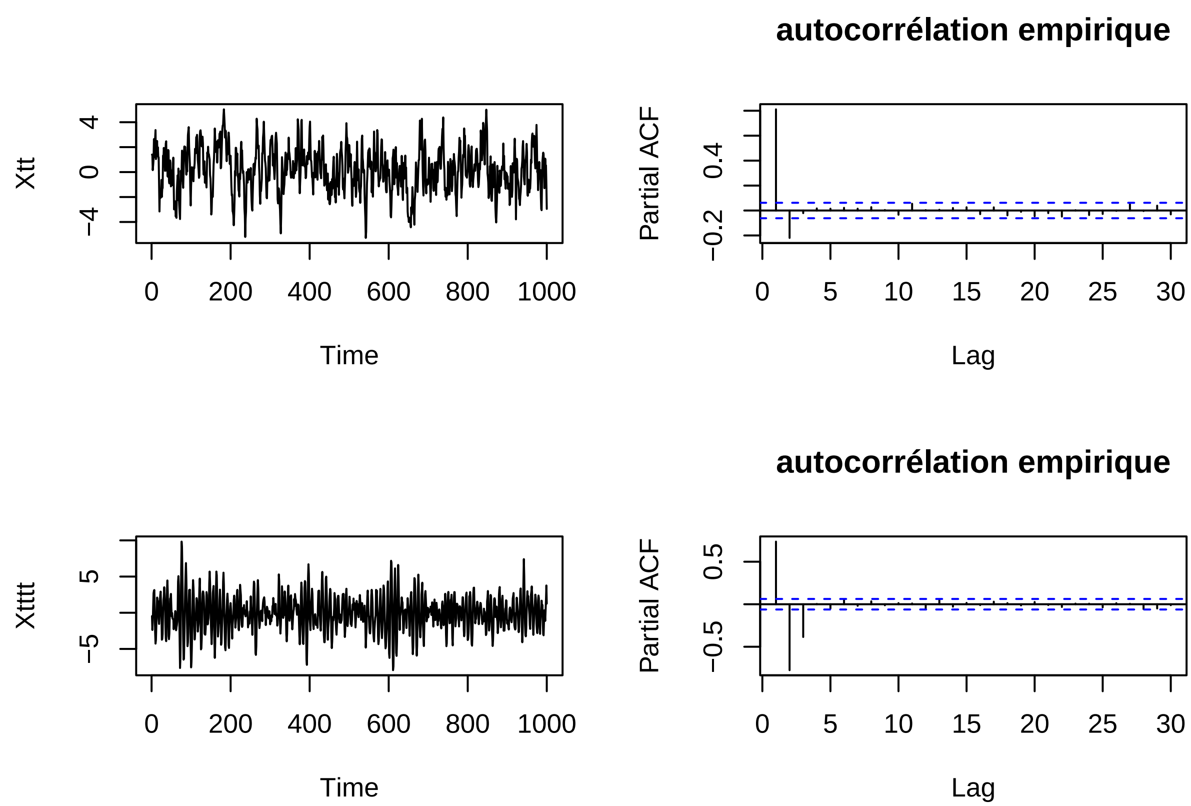
Comme il est clair que l’autocorrélation pour les processus AR ne fonctionnent pas, on teste l’autocorrélation partielle.

```
par(mfrow=c(2,2))
plot(0:10,c(1,-1/3,rep(0,9)),col=2,type='h',main="Autocorrélation partielle théorique d'un AR(1)",xlab=
pacf(Xt,main="Autocorrélation partielle empirique d'un AR(1)",lag.max=10)
plot(0:10,c(1,1,1,1,rep(0,7)),col=2,type='h',main="Autocorrélation partielle théorique d'un AR(3)",xlab=
pacf(Xttt,main="Autocorrélation partielle empirique d'un AR(3)",lag.max=10)
```



```
parametres = list(ar=c(1,-.25))
Xtt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
parametres = list(ar=c(1,-.25,-.4))
Xtttt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)

par(mfrow=c(2,2))
plot.ts(Xtt)
pacf(Xtt,main="autocorrélation empirique")
plot.ts(Xtttt)
pacf(Xtttt,main="autocorrélation empirique")
```



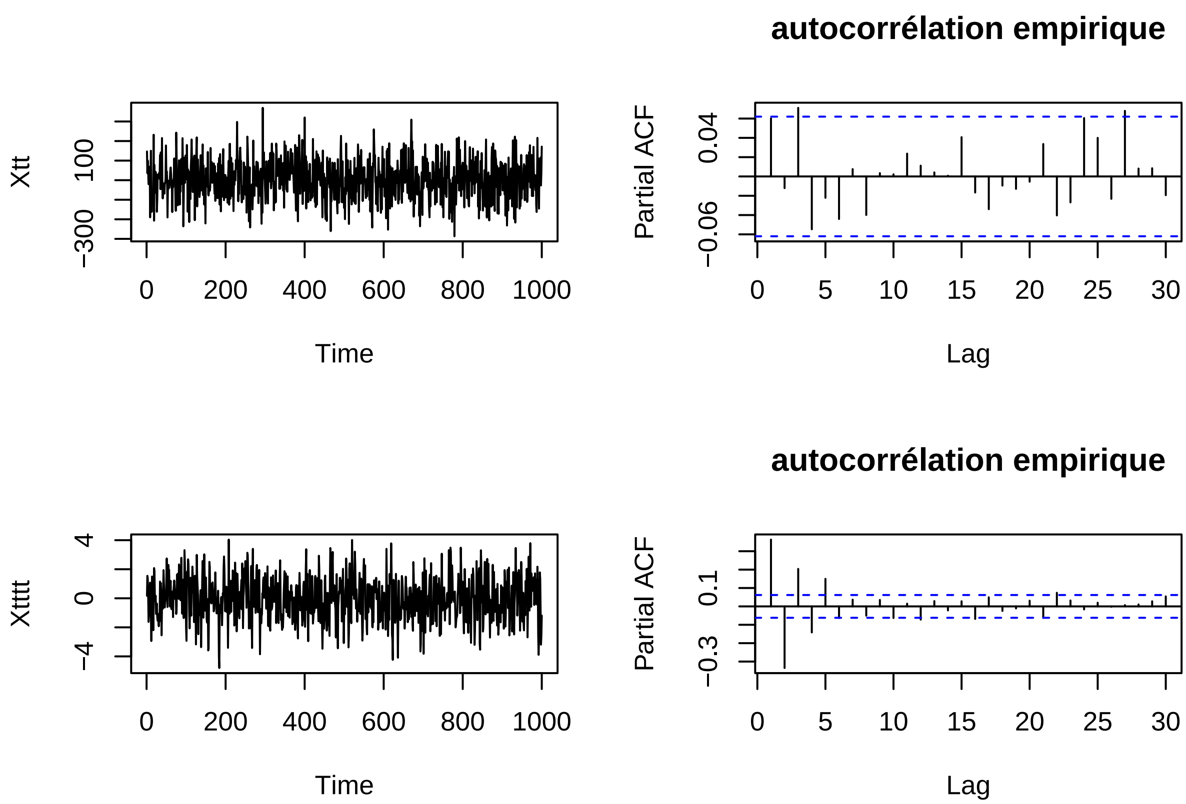
On a maintenant un bon moyen d'estimer l'ordre des processus AR. (2 et 3 ici.)

Fonction d'autocorrélation partielle d'une processus MA

Question 9:

```
parametres = list(ma=c(100,-.25))
Xtt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)
parametres = list(ma=c(1,-.25))
Xtttt=arima.sim(model=parametres,n=1000,rand.gen=rnorm)

par(mfrow=c(2,2))
plot.ts(Xtt)
pacf(Xtt,main="autocorrélation empirique")
plot.ts(Xtttt)
pacf(Xtttt,main="autocorrélation empirique")
```



On voit bien la sinusoïde amortie pour le premier MA(2) et la décroissance exponentielle pour le second. On retrouve le même problème avec les autocorrélations partielles sur les MA qu'avec les non partielles sur les AR. Les deux sont en miroir.