

UNIVERSITÉ PARIS DAUPHINE



---

# **Méthodes de Monte Carlo - PROJET -**

---

Axel GASSOT & Antoine LUCIANO

Département MIDO  
Master 1  
2020-2021

Remarque : On choisit la graine 1 pour exécuter notre code.

```
set.seed(1)
```

## Solution Exercice 1.

Soit  $f$  une densité de  $\mathbb{R}^2$  définie pour  $(x, y) \in \mathbb{R}^2$  par  $f(x, y) = a\psi(x, y)$  avec  $a \in \mathbb{R}_+^*$  et

$$\psi(x, y) = [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2 + y^4]e^{-2(x+|y|)}\mathbf{1}_{\{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}}\mathbf{1}_{\{y \in [-1, 1]\}}$$

Pour  $(X, Y)$  de densité  $f$ , l'objectif est d'estimer  $f_X$  la densité marginale de  $X$ .

### Simulation suivant la densité $f$

1.

On va montrer que pour simuler suivant  $f$ , il n'est pas nécessaire de connaître  $a$  et il suffit de trouver une constante  $m \in \mathbb{R}_+^*$  et une densité  $g$  telles que

$$\forall (x, y) \in \mathbb{R}^2, \quad \psi(x, y) \leq mg(x, y) \quad (1)$$

Or, d'après l'énoncé  $\psi(x, y) = \frac{f(x, y)}{a}$ . Alors, on remarque dans un premier temps que

$$\psi(x, y) \leq mg(x, y) \Leftrightarrow f(x, y) \leq M g(x, y) \quad \text{avec } M = am \text{ et } a \in \mathbb{R}_+^*$$

De plus, afin d'appliquer l'algorithme du rejet à  $\psi$ , on pose

$$T = \inf\{n \geq 1, U_n \leq \frac{\psi(X_n, Y_n)}{mg(X_n, Y_n)}\} \quad \text{avec } (U_n)_{n \geq 1} \stackrel{iid}{\sim} \mathcal{U}([0, 1]) \text{ et } ((X_n, Y_n))_{n \geq 1} \stackrel{iid}{\sim} g$$

On a alors, par  $\psi(x, y) = \frac{f(x, y)}{a}$ ,  $T = \inf\{n \geq 1, U_n \leq \frac{f(X_n, Y_n)}{Mg(X_n, Y_n)}\}$ . On en déduit que  $T$  est bien un temps d'arrêt pour la fonction  $f$  et donc  $(X_T, Y_T)$  suit bien la loi de densité  $f$ .

Pour conclure, l'algorithme du rejet appliqué à  $\psi$  permet bien de simuler suivant  $f$ .

Ainsi, nous cherchons  $g$  une densité de  $\mathbb{R}^2$  pour laquelle on dispose d'un générateur aléatoire telle qu'il existe une constante  $m \geq 1$  satisfaisant (1).

Tout d'abord, nous allons majorer  $C(x, y) = |\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2 + y^4$  par une constante.

On a  $\forall (x, y) \in [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-1, 1]$  :

- $|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| \leq \frac{\sqrt{2}}{2}$
- $4\cos(x)^2 \leq 4$
- $y^4 \leq 1$

Donc,  $C(x, y) \leq (5 + \frac{\sqrt{2}}{2}) \quad \forall (x, y) \in [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-1, 1]$

### Choix 1

On choisit  $g \stackrel{iid}{\sim} \mathcal{U}([-\frac{\pi}{2}, \frac{\pi}{2}]) \times \mathcal{U}([-1, 1])$  et ainsi  $g(x, y) = \frac{1}{2\pi} \mathbf{1}_{\{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}} \mathbf{1}_{\{y \in [-1, 1]\}}$ .

On a donc  $\psi(x, y) = C(x, y)e^{-2(x+|y|)}2\pi g(x, y)$

Or  $\forall (x, y) \in [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-1, 1]$  :

- $e^{-2x} \leq \exp(\pi)$

- $e^{-2|y|} \leq 1$

Ainsi, en posant  $m = (5 + \frac{\sqrt{2}}{2})e^\pi 2\pi \approx 932$  on a bien  $\psi(x, y) \leq mg(x, y)$ .

Remarque : Nous avons une trop grande valeur de  $m$  qui diminue la performance de notre simulation, ainsi nous avons préféré faire un nouveau choix pour la densité  $g$  et la constante  $m$ .

## Choix 2

On choisit  $g(x, y) = g_1(x)g_2(y)$  avec

- $g_1(x) = \frac{2e^{-2x}}{e^\pi - e^{-\pi}} \mathbf{1}_{\{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}}$  avec  $\frac{(e^\pi - e^{-\pi})}{2} = \int_{\mathbb{R}} e^{-2x} \mathbf{1}_{\{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}} dx$
- $g_2(y) = \frac{e^{-2|y|}}{1 - e^{-2}} \mathbf{1}_{\{y \in [-1, 1]\}}$  avec  $(1 - e^{-2}) = \int_{\mathbb{R}} e^{-2|y|} \mathbf{1}_{\{y \in [-1, 1]\}} dy$

On a donc  $\psi(x, y) = C(x, y)(1 - e^{-2})(e^\pi - e^{-\pi})^{\frac{1}{2}}g(x, y)$

Ainsi en posant  $m = (5 + \frac{\sqrt{2}}{2})(1 - e^{-2})(e^\pi - e^{-\pi})^{\frac{1}{2}} \approx 57$  on a bien  $\psi(x, y) \leq mg(x, y)$ .

## 2.

Nous utilisons notre **Choix 2** et donc  $g(x, y) = g_1(x)g_2(y)$  avec

- $g_1(x) = \frac{2e^{-2x}}{e^\pi - e^{-\pi}} \mathbf{1}_{\{x \in [-\frac{\pi}{2}, \frac{\pi}{2}]\}}$
- $g_2(y) = \frac{e^{-2|y|}}{1 - e^{-2}} \mathbf{1}_{\{y \in [-1, 1]\}}$

Ainsi, pour simuler suivant la densité  $g$ , nous utilisons la méthode de la fonction inverse. En effet, on note  $F_1$  et  $F_2$  les fonctions de répartition respectives de  $g_1$  et  $g_2$  et on a

$$F_1(x) = \begin{cases} 1 & \text{si } x > \frac{\pi}{2} \\ 0 & \text{si } x < -\frac{\pi}{2} \\ \frac{e^\pi - e^{-2x}}{e^\pi - e^{-\pi}} & \text{si } x \in [-\frac{\pi}{2}, \frac{\pi}{2}] \end{cases} \quad \text{et} \quad F_2(y) = \begin{cases} 1 & \text{si } y > 1 \\ 0 & \text{si } y < -1 \\ \frac{e^{2y} - e^{-2}}{2(1 - e^{-2})} & \text{si } y \in [-1, 0] \\ \frac{1 - e^{-2y}}{2(1 - e^{-2})} + \frac{1}{2} & \text{si } y \in [0, 1] \end{cases}$$

Ces deux fonctions de répartition étant continues et strictement croissantes nous avons  $F_1^{\leftarrow} = F_1^{-1}$  et  $F_2^{\leftarrow} = F_2^{-1}$  ce qui nous donne :

$$F_1^{\leftarrow}(x) = -\frac{1}{2}(\ln(1 - x(1 - e^{-2\pi})) + \pi) \quad \text{et} \quad F_2^{\leftarrow}(y) = \left(\frac{\ln(2y(1 - e^{-2}))}{2} - 1\right)\mathbf{1}_{y \in [0, \frac{1}{2}]} - \frac{\ln(2y - 1)(1 - e^{-2})}{2}\mathbf{1}_{y \in [\frac{1}{2}, 1]}$$

```
# Simulation selon la densité g1 avec la methode de la fonction inverse
fct_inv_x <- function(x){
  return (-(log(1-x*(1-exp(-2*pi))))+pi)/2)
}

# Simulation selon la densité g2 avec la methode de la fonction inverse
fct_inv_y <- function(x){
  return ((log(2*x*(exp(2)-1)+1)-2)/2*(x<1/2)-(log(1-(2*x-1)*(1-exp(-2)))/2*(x>=1/2)))
}
```

```

# Simulation de n réalisations suivant la densité g
rgen_g <- function(n) {
  u <- runif(n)
  x <- fct_inv_x(u)
  y <- fct_inv_y(u)
  return(cbind(x,y))
}

# Constante m
m <- ((5+sqrt(2)/2)*(1-exp(-2))*(exp(pi)-exp(-pi)))/2

# Fonction rho
rho<- function(x,y){
  return ((abs(sin((2/pi)*x^2-pi/4))+4*cos(x)^2+y^4)/(5+sqrt(2)/2))
}

```

### Algorithme de rejet

Pour avoir un algorithme de rejet le plus efficace possible, nous calculons à chaque itération la moyenne des valeurs prises par  $\rho$  qu'on appelle ici *ratio*.

*ratio* est le taux d'acceptation de notre densité  $g$  et  $\frac{1}{ratio}$  correspond donc au nombre moyen d'essai moyen pour qu'une valeur soit acceptée.

Ainsi, nous pouvons adapter le nombre de simulation de  $g$  à l'itération suivante en simulant  $k/ratio$  où  $k$  correspond au nombre de simulations manquantes.

```

# Simulation de n réalisations suivant la densité f par methode du rejet
rgen_f <- function(n){
  ans <- c() #Echantillon de sortie
  rho.ans <- c()
  nl <- 0
  nt <- 0
  ratio <- 1
  k <- n
  while (k>0) {
    nl <- nl + 1
    nt <- nt + floor(k/ratio)
    sim_g <- rgen_g(floor(k/ratio))
    x <- sim_g[,1]
    y <- sim_g[,2]
    rho <- rho(x,y)
    w <- which(runif(floor(k/ratio)) <= rho)
    rho.ans <- append(rho.ans,rho)
    ans <- rbind(ans, sim_g[w, ])
    ratio <- mean(rho)
    k <- n - nrow(ans)
  }
  return(list("Simulation"=ans[1:n,], "Rho"=rho.ans, "nl"=nl, "nt"=nt))
}

```

Nous allons à présent auto-évaluer notre simulation.

```
# Autoévaluation

n <- 10000

n_test <- 100

nl <- 0
nt <- 0

for (i in 1:n_test){
  test <- rgen_f(n)
  nl <- nl + test$nl
  nt <- nt + test$nt
}

nl <- nl/n_test
nt <- nt/n_test
```

L'auto-évaluation de de notre simulation nous donne :

- $nt = 3.847232 \times 10^4$
- $nl = 2.87$

où  $n_t$  désigne le nombre moyen de simulations suivant  $g$  et  $n_l$  le nombre moyen de passages dans une boucle *for* ou *while* pour le code utilisé pour générer les réalisations de  $f$ .

Nous sommes donc satisfaits de notre simulation.

### 3.

```
n <- 10000

# Echantillon z de densité f
f <- rgen_f(n)
z <- f$Simulation
```

## Méthode n°1 – Estimation de $a$

### 4.(a)

Nous avons dans un premier temps

$$\mathbb{E}_g[\rho(X, Y)] = \int_{\mathbb{R}^2} \rho(x, y) g(x, y) \, dx \, dy = \int_{\mathbb{R}^2} \frac{\psi(x, y)}{mg(x, y)} g(x, y) \, dx \, dy = \frac{1}{m} \int_{\mathbb{R}^2} \psi(x, y) \, dx \, dy$$

Or,  $f(x, y) = a\psi(x, y)$  où  $f$  est une densité d'où

$$\int_{\mathbb{R}^2} f(x, y) \, dx \, dy = 1 \iff \int_{\mathbb{R}^2} \psi(x, y) \, dx \, dy = \frac{1}{a}$$

Ainsi

$$\mathbb{E}[\rho(X, Y)] = \frac{1}{ma} \text{ avec } (X, Y) \sim g$$

On en déduit l'estimateur classique de Monte Carlo de  $\frac{1}{a}$ ,

$$\widehat{\delta}_n = \frac{1}{n} \sum_{i=1}^n m\rho(X_i, Y_i) \quad \text{avec } (X_i, Y_i) \stackrel{iid}{\sim} g$$

Donc nous prenons

$$\widehat{b}_n = \frac{1}{\widehat{\delta}_n} = \frac{1}{\frac{1}{n} \sum_{i=1}^n m\rho(X_i, Y_i)} \quad \text{avec } (X_i, Y_i) \stackrel{iid}{\sim} g$$

**Biais de l'estimateur.** Comme la fonction  $x \mapsto \frac{1}{x}$  est convexe sur  $\mathbb{R}_+^*$ , par l'inégalité de Jensen :

$$\mathbb{E}[\widehat{b}_n] = \mathbb{E}\left[\frac{1}{\widehat{\delta}_n}\right] \geq \frac{1}{\mathbb{E}[\widehat{\delta}_n]}$$

De plus, les variables aléatoires  $(m\rho(X_n, Y_n))_{n \geq 1}$  étant identiquement distribuées, on a :

$$\mathbb{E}[\widehat{\delta}_n] = \mathbb{E}_g[m\rho(X_1, Y_1)] = \frac{1}{a}$$

Donc  $\mathbb{E}[\widehat{b}_n] \geq a$ , on obtient que l'estimateur est biaisé.

**Convergence de l'estimateur.** Les variables aléatoires  $(m\rho(X_n, Y_n))_{n \geq 1}$  sont *i.i.d.* et d'espérance finie sous  $g$ . La loi forte des grands nombres donne  $\widehat{\delta}_n \xrightarrow[n \rightarrow +\infty]{p.s.} \mathbb{E}_g[m\rho(X_1, Y_1)] = \frac{1}{a}$

D'où, par continuité de  $x \mapsto \frac{1}{x}$ , on a  $\widehat{b}_n \xrightarrow[n \rightarrow +\infty]{p.s.} a$

**Intervalle de confiance.** Les variables aléatoires  $(m\rho(X_n, Y_n))_{n \geq 1}$  sont *i.i.d.* et de variance finie,  $m\rho$  étant de carré intégrable par rapport à  $g$ . En effet,

$$\begin{aligned} \mathbb{E}_g[m^2\rho(X, Y)^2] &= \int_{\mathbb{R}^2} \rho(x, y)^2 g(x, y) \, dx \, dy \\ &= \int_{\mathbb{R}^2} \frac{\psi(x, y)^2}{g(x, y)} \, dx \, dy \\ &\leq \int_{\mathbb{R}^2} m\psi(x, y) \, dx \, dy \quad \text{car } \psi(x, y) \leq mg(x, y) \iff \frac{\psi(x, y)}{g(x, y)} \leq m \\ &\leq \frac{m}{a} < +\infty \end{aligned}$$

Le théorème centrale limite donne

$$\sqrt{n}(\hat{\delta}_n - \frac{1}{a}) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma^2)$$

D'après la Méthode Delta pour  $h : x \mapsto \frac{1}{x}$ , on obtient

$$\sqrt{n}(\hat{b}_n - a) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma^2 a^4) \text{ car } h'(x) = -\frac{1}{x^2}$$

On en déduit l'intervalle de confiance de  $a$  au niveau de confiance  $1 - \alpha$ ,

$$\begin{aligned} IC_{1-\alpha} &= [\hat{b}_n - q_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2 a^4}{n}}, \hat{b}_n + q_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2 a^4}{n}}] \\ &= [\hat{b}_n - q_{1-\frac{\alpha}{2}} \sqrt{Var[\hat{b}_n] a^4}, \hat{b}_n + q_{1-\frac{\alpha}{2}} \sqrt{Var[\hat{b}_n] a^4}] \end{aligned}$$

où  $q_{1-\frac{\alpha}{2}}$  est le quantile d'ordre  $1 - \frac{\alpha}{2}$  de la loi normale centrée réduite. Dans la pratique, on estime la variance  $\sigma^2$  via la variance empirique associée aux réalisations de la variable aléatoire  $(m\rho(X_n, Y_n))_{n \geq 1}$  :

$$\hat{\sigma}_n^2 = \frac{1}{n} \sum_{k=1}^n \{m\rho(X_k, Y_k) - \hat{b}_n\}^2$$

Et on estime  $a^4$  avec  $\hat{b}_n^4$ , par la loi forte des grands nombres et le théorème de Slutsky. Finalement,

$$IC_{1-\alpha} = [\hat{b}_n - q_{1-\frac{\alpha}{2}} \hat{b}_n^2 \sqrt{\frac{\hat{\sigma}_n^2}{n}}, \hat{b}_n + q_{1-\frac{\alpha}{2}} \hat{b}_n^2 \sqrt{\frac{\hat{\sigma}_n^2}{n}}]$$

#### 4.(b)

```
# Estimateur b_n et intervalle de confiance associé
bn_estim <- function(y, level) {
  delta <- 1/mean(y)
  s2 <- var(y)*(delta^4)
  eps <- (pnorm(0.5 * (1 + level)) * sqrt(s2/length(y)))
  return(data.frame(
    value = delta,
    var = s2,
    ic_inf = delta - eps,
    ic_sup = delta + eps
  ))
}

# Variable dans l'estimateur b_n
y_b <- m*f$Rho

b_n <- bn_estim(y_b, 0.95)
```

Nous avons donc  $\hat{b}_n = 0.0669757$

Et l'intervalle de confiance asymptotique associé de  $a$  au niveau 95% est

$$[0.066725, 0.0672264]$$

#### 4.(c)

Afin d'estimer le biais de l'estimateur  $\hat{b}_n$  nous utilisons une méthode *Bootsrtap*.

En effet, on génère à partir de  $K$  échantillons de taille 1000, aléatoirement tirés parmi l'échantillon des valeurs de  $\rho(x, y)$ ,  $K$  estimateurs afin d'estimer l'espérance de  $\hat{b}_n$  par la méthode de Monte Carlo classique :

$$\frac{1}{K} \sum_{k=1}^K \hat{b}_n^{(k)}.$$

Notre estimation du biais est alors

$$\widehat{B}(\hat{b}_n, a) = \left( \frac{1}{K} \sum_{k=1}^K \hat{b}_n^{(k)} \right) - \hat{b}_n$$

```
# Methode Bootstrap

# On génère à partir de k échantillons de taille 10000 k estimateurs afin
# d'estimer l'esperance de l'estimateur

k <- 10000 #nombre d'estimateurs
v_boot <- c()
for (i in 1:k) {
  # x_boot : vecteur aleatoire de taille 10000 des valeurs de rho
  x_boot <- sample(x = m*f$Rho, size = 1000, replace = TRUE)
  # b_n_boot : estimateur de x_boot
  b_n_boot <- bn_estim(x_boot, 0.95)
  # v_boot : vecteur de k estimateurs
  v_boot <- append(v_boot, b_n_boot$value) #vecteur de k estimateurs
}

# Calcul du biais estimé
biais_estim <- (mean(v_boot) - b_n$value)
```

La valeur du biais estimé est  $7.8993464 \times 10^{-5}$  ce qui nous semble assez faible. L'estimateur est donc biaisé mais reste correct.

### 5.(a)

Nous avons dans un premier temps

$$\begin{aligned} \mathbb{E}_f \left[ \frac{g(X,Y)}{\psi(X,Y)} \right] &= \int_{\mathbb{R}^2} \frac{g(x,y)}{\psi(x,y)} f(x,y) dx dy \\ &= \int_{\mathbb{R}^2} g(x,y) a \, dx \, dy \quad \text{car } \frac{f(x,y)}{\psi(x,y)} = a \\ &= a \quad \text{car } g \text{ est une densité donc } \int_{\mathbb{R}^2} g(x,y) dx dy = 1 \end{aligned}$$

De plus,  $\frac{g(X,Y)}{\psi(X,Y)} = \frac{1}{m\rho(X,Y)}$  et ainsi  $\mathbb{E}_f \left[ \frac{1}{m\rho(X,Y)} \right] = a$ .

On en déduit l'estimateur classique de Monte Carlo de a :

$$\widehat{a}_n = \frac{1}{n} \sum_{i=1}^n \frac{1}{m\rho(X_i, Y_i)} \quad \text{avec } (X_i, Y_i) \stackrel{iid}{\sim} f$$

**Biais de l'estimateur.** Les variables aléatoires  $(\frac{1}{m\rho(X_n, Y_n)})_{n \geq 1}$  étant identiquement distribuées, on a :

$$\mathbb{E}[\widehat{a}_n] = \mathbb{E}_f \left[ \frac{1}{m\rho(X_1, Y_1)} \right] = a$$

L'estimateur est donc sans biais.



**Convergence de l'estimateur.** Les variables aléatoires  $(\frac{1}{m\rho(X_n, Y_n)})_{n \geq 1}$  sont *i.i.d.* et d'espérance finie sous  $f$ . La loi forte des grands nombres donne  $\hat{a}_n \xrightarrow[n \rightarrow +\infty]{p.s.} \mathbb{E}_f[\frac{1}{m\rho(X_1, Y_1)}] = a$ .

Donc  $\hat{a}_n \xrightarrow[n \rightarrow +\infty]{p.s.} a$ .

**Intervalle de confiance.** Les variables aléatoires  $(\frac{1}{m\rho(X_n, Y_n)})_{n \geq 1}$  sont *i.i.d.* et de variance finie. Le théorème centrale limite donne :

$$\sqrt{n}(\hat{a}_n - a) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma^2) \text{ avec } \sigma^2 = \text{Var}[\frac{1}{m\rho(X_1, Y_1)}]$$

On en déduit l'intervalle de confiance de  $a$  au niveau de confiance  $1 - \alpha$ ,

$$\begin{aligned} IC_{1-\alpha} &= [\hat{a}_n - q_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}}, \hat{a}_n + q_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}}] \\ &= [\hat{a}_n - q_{1-\frac{\alpha}{2}} \sqrt{\text{Var}[\hat{a}_n]}, \hat{a}_n + q_{1-\frac{\alpha}{2}} \sqrt{\text{Var}[\hat{a}_n]}] \end{aligned}$$

où  $q_{1-\frac{\alpha}{2}}$  est le quantile d'ordre  $1 - \frac{\alpha}{2}$  de la loi normale centrée réduite. Dans la pratique, on estime la variance  $\sigma^2$  via la variance empirique associée aux réalisations de la variable aléatoire  $(\frac{1}{m\rho(X_n, Y_n)})_{n \geq 1}$  :

$$\hat{\sigma}_n^2 = \frac{1}{n} \sum_{k=1}^n \left\{ \frac{1}{m\rho(X_k, Y_k)} - \hat{a}_n \right\}^2$$

## 5.(b)

```
# Estimateur a_n et intervalle de confiance associé
an_estim <- function(y, level) {
  delta <- mean(y)
  s2 <- var(y)
  eps <- (pnorm(0.5 * (1 + level)) * sqrt(s2 / length(y)))
  return(data.frame(
    value = delta,
    var = s2,
    ic_inf = delta - eps,
    ic_sup = delta + eps
  ))
}

# Variable dans l'estimateur a_n
y_a <- function(x,y){
  return((1/(m*rho(x,y))))
}

a_n <- an_estim(y_a(z[,1],z[,2]), 0.95)
```

‘ Nous avons donc  $\hat{a}_n = 0.0668145$

Et l'intervalle de confiance asymptotique associé de  $a$  au niveau 95% est

$$[0.06636, 0.0672691]$$

## 6.

Afin de calculer le rapport des coûts pour lesquels  $\hat{a}_n$  et  $\hat{b}_n$  atteignent la même précision nous calculons le rapport des variances de  $\hat{a}_n$  et  $\hat{b}_n$  calculés précédemment. On a

$$\frac{\text{Var}[\widehat{a}_n]}{\text{Var}[\widehat{b}_n]} = \frac{\text{Var}_f[\frac{1}{m\rho(X,Y)}]}{\text{Var}_g[m\rho(X,Y)]\widehat{b}_n^4}$$

*# Rapport des coûts*

```
st_1 <- a_n$var/b_n$var
```

On voit alors que  $\frac{\text{Var}[\widehat{a}_n]}{\text{Var}[\widehat{b}_n]} = 0.857052 < 1 \Rightarrow \text{Var}[\widehat{a}_n] < \text{Var}[\widehat{b}_n]$

On en déduit alors que  $\widehat{a}_n$  est plus efficace que  $\widehat{b}_n$  pour l'estimation de  $a$ .

### 7.(a)

Soit  $x \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ , alors

$$\psi(x, y) = [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2 + y^4]e^{-2(x+|y|)}\mathbf{1}_{\{y \in [-1, 1]\}}$$

De plus,  $f(x, y) = a\psi(x, y) \Rightarrow f_X(x) = a \int_{\mathbb{R}} \psi(x, y) dy$

$$\begin{aligned} \text{Et } \int_{\mathbb{R}} \psi(x, y) dy &= \int_{[-1, 1]} [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2 + y^4]e^{-2(x+|y|)} dy \\ &= e^{-2x} [ [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2] \frac{e^2-1}{e^2} + \int_{[-1, 1]} y^4 e^{-2|y|} dy ] \end{aligned}$$

Avec  $\int_{[-1, 1]} y^4 e^{-2|y|} dy = 2 \int_{[-1, 0]} y^4 e^{2y} dy = \frac{3(e^2-7)}{4e^2}$  par des intégrations par parties successives.

Ainsi,

$$\int_{\mathbb{R}} \psi(x, y) dy = e^{-2x} [ [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2] \frac{e^2-1}{e^2} + \frac{3(e^2-7)}{4e^2} ]$$

On en déduit alors la densité marginale de X

$$f_X(x) = a.e^{-2x} [ [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2] \frac{e^2-1}{e^2} + \frac{3(e^2-7)}{4e^2} ]$$

Finalement, par le théorème de Slutsky, comme  $\widehat{a}_n \xrightarrow[n \rightarrow +\infty]{p.s.} a$ , on prend

$$\boxed{\widehat{f}_{X,n}(x) = \widehat{a}_n . e^{-2x} [ [|\sin(\frac{2}{\pi}x^2 - \frac{\pi}{4})| + 4\cos(x)^2] \frac{e^2-1}{e^2} + \frac{3(e^2-7)}{4e^2} ]} \quad \text{avec } \widehat{f}_{X,n}(x) \xrightarrow[n \rightarrow +\infty]{p.s.} f_X(x)$$

(Nous aurions pu prendre  $\widehat{b}_n$  plutôt que  $\widehat{a}_n$  mais nous avons vu précédemment que  $\widehat{a}_n$  est plus efficace.)

### 7.(b)

```
psi_x <- function(x){
  return(exp(-2*x)*(((exp(2)-1)/exp(2))*(abs(sin((2/pi)*x*x-(pi/4)))+4*cos(x)*cos(x))+3*(exp(2)-7)/(4*exp(2))))
}
```

*# Estimateur  $f_X(x)$  de la densité marginale de X*

```
f_x_n <- function(a, x) {
  return(a*psi_x(x))
}
```

*# Intervalle d'évaluation*

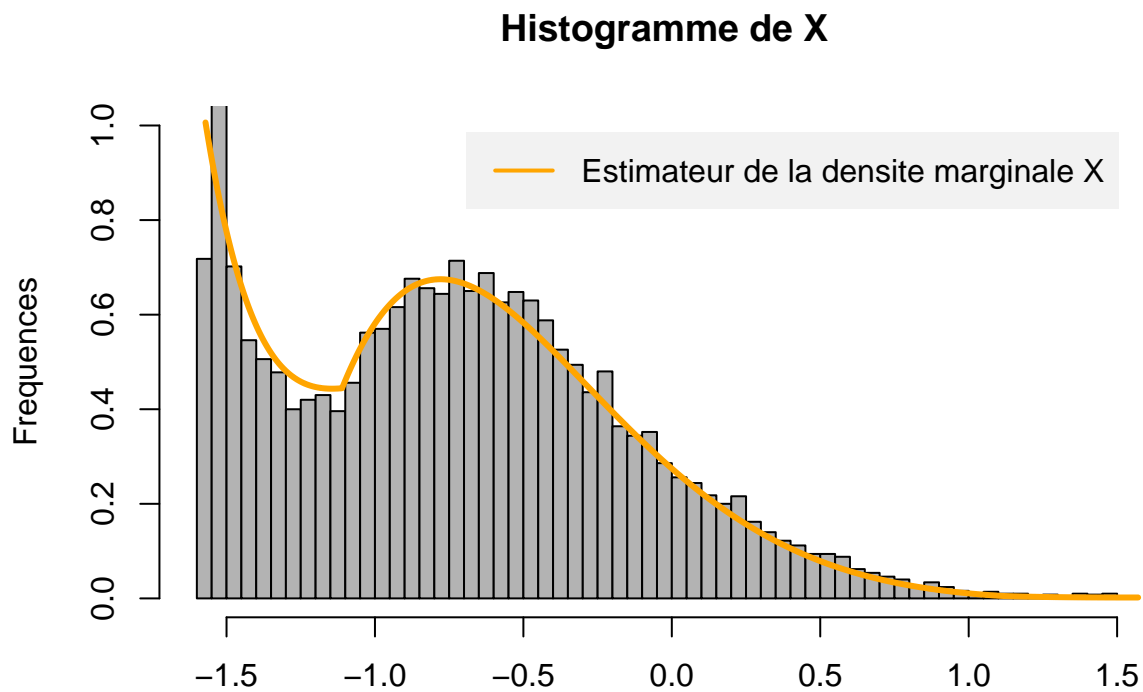
```
s <- seq((-pi/2), (pi/2), length.out = 10000)

# Comparaison graphique de la distribution marginale de l'échantillon z à l'estimateur f_x_n
par(mfrow = c(1, 1))

hist(
  x = z[, 1], freq = F, main = "Histogramme de X", xlab = "", ylab = "Frequencies",
  col = "grey70", breaks = 60, ylim = c(0,1))

lines( s, f_x_n(a_n$value, s), col = "orange", lwd = 3)

legend("topright", "Estimateur de la densité marginale X", col = "orange",
      lty = c(1, 4), lwd = c(2, 3), box.lty = 0, bg = "gray95", inset = .05
    )
```



## Méthode n°2 – Estimateur ponctuel

8.

Soient  $(X_1, Y_1), \dots, (X_n, Y_n)$  une suite de variables indépendantes suivant la loi jointe  $f_{X,Y}(x, y)$  et  $\omega(\cdot)$  une densité quelconque.

On note  $\hat{\omega}_n(t) = \frac{1}{n} \sum_{k=1}^n \frac{\psi(t, Y_k) \omega(X_k)}{\psi(X_k, Y_k)}$

La loi forte des grands nombres pour la suite de variables aléatoires *i.i.d.*  $(\frac{\psi(t, Y_n) \omega(X_n)}{\psi(X_n, Y_n)})_{n \geq 1}$  nous donne

$$\hat{\omega}_n(t) \xrightarrow[n \rightarrow +\infty]{p.s.} \mathbb{E}\left[\frac{\psi(t, Y_1) \omega(X_1)}{\psi(X_1, Y_1)}\right]$$

De plus,

$$\begin{aligned} \mathbb{E}\left[\frac{\psi(t, Y_1) \omega(X_1)}{\psi(X_1, Y_1)}\right] &= \int_{\text{supp}(f_{X,Y})} \frac{\psi(t, y) \omega(x)}{\psi(x, y)} f_{X,Y}(x, y) \, dx dy \\ &= \int_{\text{supp}(f_{X,Y})} \frac{a f_{X,Y}(t, y) \omega(x)}{a f_{X,Y}(x, y)} f_{X,Y}(x, y) \, dx dy \quad \text{car } \psi(x, y) = \frac{f_{X,Y}(x, y)}{a} \\ &= \int_{\text{supp}(f_{X,Y})} \omega(x) f_{X,Y}(x, y) \, dx dy \\ &= \int_{\text{supp}(f_Y)} f_{X,Y}(x, y) \underbrace{\left( \int_{\text{supp}(f_X)} \omega(x) \, dx \right)}_{=1} dy \quad \text{d'après le théorème de Fubini} \\ &= \int_{\text{supp}(f_Y)} f_{X,Y}(x, y) \, dy \quad \text{car } \omega(\cdot) \text{ est une densité et } \text{supp}(\omega) \subseteq \text{supp}(f_x) \\ &= f_X(t) \end{aligned}$$

Ainsi, on a bien montré que

$$\boxed{\hat{\omega}_n(t) \xrightarrow[n \rightarrow +\infty]{p.s.} f_X(t)}$$

**Intervalle de confiance.** Les variables aléatoires  $(\frac{\psi(t, Y_n) \omega(X_n)}{\psi(X_n, Y_n)})_{n \geq 1}$  sont *i.i.d.* et de variance finie. Le théorème central limite donne

$$\sqrt{n}(\hat{\omega}_n(t) - f_X(t)) \xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma^2) \quad \text{avec } \sigma^2 = \text{Var}\left[\frac{\psi(t, Y_1) \omega(X_1)}{\psi(X_1, Y_1)}\right]$$

On en déduit l'intervalle de confiance de  $f_X(t)$  au niveau de confiance  $1 - \alpha$ ,

$$\begin{aligned} IC_{1-\alpha} &= [\hat{\omega}_n(t) - q_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}}, \hat{\omega}_n(t) + q_{1-\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{n}}] \\ &= [\hat{\omega}_n(t) - q_{1-\frac{\alpha}{2}} \sqrt{\text{Var}[\hat{\omega}_n(t)]}, \hat{\omega}_n(t) + q_{1-\frac{\alpha}{2}} \sqrt{\text{Var}[\hat{\omega}_n(t)]}] \end{aligned}$$

où  $q_{1-\frac{\alpha}{2}}$  est le quantile d'ordre  $1 - \frac{\alpha}{2}$  de la loi normale centrée réduite. Dans la pratique, on estime la variance  $\sigma^2$  via la variance empirique associée aux réalisations de la variable aléatoire  $(\frac{\psi(t, Y_n) \omega(X_n)}{\psi(X_n, Y_n)})_{n \geq 1}$  :

$$\hat{\sigma}_n^2(t) = \frac{1}{n} \sum_{k=1}^n \left\{ \frac{\psi(t, Y_k) \omega(X_k)}{\psi(X_k, Y_k)} - \hat{\omega}_n(t) \right\}^2$$

## 9.

On sait que  $\psi(x, y) = \frac{f(x, y)}{a}$ . De plus, les variables aléatoires  $X$  et  $Y$  sont indépendantes donc  $f_{X,Y}(x, y) = f_X(x)f_Y(y)$ . Ainsi, on a

$$\hat{\omega}_n(t) = \frac{1}{n} \sum_{k=1}^n \frac{f_X(t)f_Y(Y_k)\omega(X_k)}{f_X(X_k)f_Y(Y_k)} = \frac{1}{n} \sum_{k=1}^n \frac{f_X(t)\omega(X_k)}{f_X(X_k)}$$

Nous cherchons la densité  $\omega(\cdot)$  qui minimise la variance de  $\hat{\omega}_n(t)$ . Or

$$\begin{aligned} \min_{\omega(\cdot)} (Var[\hat{\omega}_n(t)]) &\iff \min_{\omega(\cdot)} \left( \frac{f_X(t)^2}{n} Var\left[\frac{\omega(X_1)}{f_X(X_1)}\right] \right) && \text{car les v.a. } \left(\frac{\omega(X_n)}{f_X(X_n)}\right)_{n \geq 1} \text{ sont } i.i.d. \\ &\iff \min_{\omega(\cdot)} (\mathbb{E}[(\frac{f_X(t)\omega(X_1)}{f_X(X_1)})^2] - \mathbb{E}[(\frac{f_X(t)\omega(X_1)}{f_X(X_1)})]^2) \\ &\iff \min_{\omega(\cdot)} \left( \int_{supp(f_X)} \frac{\omega(x)^2}{f_X(x)} dx - \underbrace{\left[ \int_{supp(f_X)} \omega(x) dx \right]^2}_{=1} \right) \\ &\iff \min_{\omega(\cdot)} \left( \int_{supp(f_X)} \frac{\omega(x)^2}{f_X(x)} dx - 1 \right) \end{aligned}$$

$$\text{Or } \left( \int_{supp(f_X)} \frac{\omega(x)^2}{f_X(x)} dx - 1 \right) = Var\left[\frac{\omega(X_1)}{f_X(X_1)}\right] \geq 0$$

Ainsi chercher une densité  $\omega(\cdot)$  qui minimise la variance de  $\hat{\omega}_n(t)$  revient à chercher une densité  $\omega(\cdot)$  pour laquelle  $\int_{supp(f_X)} \frac{\omega(x)^2}{f_X(x)} dx = 1$ .

On remarque alors qu'un choix évident est :

$$\boxed{\omega(\cdot) = f_X(\cdot)}$$

Cependant,  $f_X$  est la densité que nous cherchons à estimer, nous ne pouvons donc l'utiliser.

Dans la pratique, nous utiliserons alors l'estimateur de  $f_X$  construit précédemment, à savoir  $\hat{f}_{X,n}(x)$  qui converge presque-sûrement vers  $f_X$ .

## 10.

```
# Estimateur w_n et intervalle de confiance associé?
wn_estim <- function(t, y, level) {
  delta <- mean(y(t))
  s2 <- var(y(t))
  eps <- (pnorm(0.5 * (1 + level)) * sqrt(s2 / length(y(t))))
  return(data.frame(
    value = delta,
    var = s2,
    ic_inf = delta - eps,
    ic_sup = delta + eps
  ))
}

# Fonction psi
psi <- function(x, y){
```

```

    return(((abs(sin((2/pi)*x-x-pi/4))+4*cos(x)*cos(x)+y^4)*exp(-2*(x+abs(y))))*(abs(x) <= (pi/2))*(abs(y)
  }

# Variable dans l'estimateur w_n
y_w <- function(t){
  return(((psi(t, z[, 2])*f_x_n(a_n$value,z[, 1]))/(psi(z[, 1], z[, 2]))))
}

w_n <- wn_estim(-1, y_w, 0.95)

```

Nous avons donc  $\hat{w}_n(-1) = 0.6016734$

Et l'intervalle de confiance asymptotique associé de  $a$  au niveau 95% est

$$[0.6012251, 0.6021217]$$

## 11.

Afin de calculer le rapport des coûts pour lesquels  $\hat{w}_n(-1)$  et  $\hat{f}_{X,n}(-1)$  atteignent la même précision nous calculons le rapport des variances de  $\hat{w}_n(-1)$  et  $\hat{f}_{X,n}(-1)$ . On a

$$\frac{Var[\hat{w}_n(-1)]}{Var[\hat{f}_{X,n}(-1)]} = \frac{Var_f[\frac{\psi(-1,Y)\omega(X)}{\psi(X,Y)}]}{Var_g[\hat{f}_{X,n}(-1)]}$$

```

# Rapport des couts
st_2 <- w_n$var/var(y_a(z[,1],z[,2])*psi_x(-1)^2)

```

On voit alors que  $\frac{Var[\hat{w}_n(-1)]}{Var[\hat{f}_{X,n}(-1)]} = 1.6982642 \times 10^{-4} < 1 \Rightarrow Var[\hat{w}_n(-1)] < Var[\hat{f}_{X,n}(-1)]$

On en déduit alors que  $\hat{w}_n(-1)$  est beaucoup plus efficace que  $\hat{f}_{X,n}(-1)$  pour l'estimation de  $f_X$ .

## Solution Exercice 2.

Soit  $X = (X_1, X_2, X_3)$  un vecteur aléatoire de  $\mathbb{R}^3$  distribué suivant la loi  $\mathcal{N}(\mu, \Sigma)$  avec

$$\mu = \begin{pmatrix} 0.1 \\ 0 \\ 0.1 \end{pmatrix} \text{ et } \Sigma = \begin{pmatrix} 0.047 & 0 & 0.0117 \\ 0 & 0.047 & 0 \\ 0.0117 & 0 & 0.047 \end{pmatrix}$$

On s'intéresse à

$$\delta = \mathbb{E}[\min(3, \frac{1}{3} \sum_{i=1}^n e^{-X_k})]$$

1.

Pour simuler la loi de  $X$  on code une fonction `rmvnorm(n, mu, sigma)` qui permet de générer  $n$  simulations de la loi normale multivariée de moyenne  $\mu$  et de matrice de variance-covariance  $\Sigma$ .

Cette fonction repose sur la proposition suivante du cours :

Soit  $Z \sim \mathcal{N}(0_3, I_3)$ ,

on utilise la décomposition de Cholesky de  $\Sigma$  i.e.  $\Sigma = LL^T$  avec  $L$  une matrice triangulaire inférieure.

Ainsi  $X = \mu + LZ$  avec  $\mu \in \mathbb{R}^3$  suit la loi  $\mathcal{N}(\mu, \Sigma)$

```
rmvnorm <- function(n, mu, sigma) { # Simulation de X
  Z <- matrix(rnorm(3 * n), nrow = 3, ncol = n)
  X <- t(chol(sigma))
  return(mu + X %*% Z)
}

# On utilise les données de l'exercice pour simuler 10000 simulations de X
n <- 10000
mu <- c(0.1, 0, 0.1)
sigma <- matrix(c(0.047, 0, 0.0117, 0, 0.047, 0, 0.0117, 0, 0.047), 3, 3)

x <- rmvnorm(n, mu, sigma)
```

2.(a)

$$\delta = \mathbb{E}[\min(3, \frac{1}{3} \sum_{i=1}^n e^{-X_k})].$$

On calcule l'estimateur de Monte-Carlo classique en simulant  $n$  variables aléatoires  $(X_i)_{i \in [1, n]} = ((X_{1,i}, X_{2,i}, X_{3,i}))_{i \in [1, n]}$ , *i.i.d.* suivant la loi de  $X$ .

$$\bar{\delta}_n = \frac{1}{n} \sum_{i=1}^n \min(3, \frac{1}{3} \sum_{k=1}^3 e^{-X_{k,i}})$$

2.(b)

On utilise les  $n$  simulations de la question 1.

On code une fonction  $h(x, n)$  qui prend en arguments un vecteur  $(X_i)_{i \in [1, n]}$  de taille  $n$  et renvoie un vecteur  $(h(X_i))_{i \in [1, n]}$  où  $h(X) = \min(3, \frac{1}{3} \sum_{k=1}^3 e^{-X_k})$ .

On utilise la fonction *pmin*, qui nous permet de retourner le minimum de deux vecteurs, afin d'optimiser l'efficacité de la fonction.

Pour calculer l'erreur quadratique moyenne, on utilise la formule  $\frac{\text{Var}[h(X)]}{n}$  et la fonction *var(.)* qui retourne la variance empirique.

```
h <- function(x, n) { # Fonction h utilisant pmin pour prendre le min de 2 vecteurs
  return(pmin(rep(3, n), colMeans(exp(-x))))
}
```

```
h.x <- h(x, n)
delta <- mean(h.x)
delta
```

```
## [1] 0.9591481
```

```
var.h <- var(h.x)
erreur <- var.h / n
erreur
```

```
## [1] 1.712204e-06
```

Ici on a  $\bar{\delta}_n = 0.9591481$  et l'erreur quadratique vaut  $1.712204 \times 10^{-6}$ .

### 3.(a)

On prend  $A(X) = 2\mu - X$  qui est mesurable. Ainsi,  $A(X)$  suit la même loi que  $X$  comme le confirme les deux graphes ci-dessous.

```
# On introduit la variable antithétique A= 2*mu - X
a <- -x + 2 * mu
```

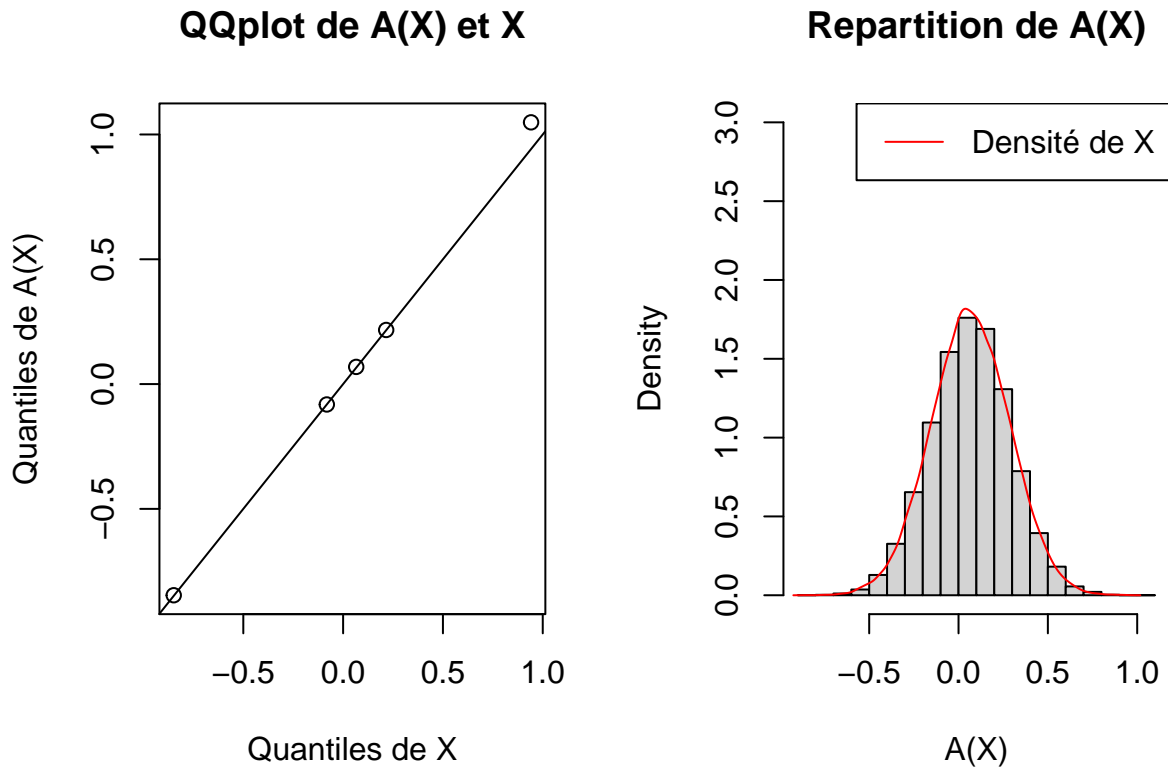
```
par(mfrow = c(1, 2))
```

```
q_a <- quantile(a)
q_x <- quantile(x)
```

```
qqplot(q_x, q_a, xlab = "Quantiles de X", ylab = "Quantiles de A(X)", main = "QQplot de A(X) et X")
abline(a = 0, b = 1)
```

```
hist(a, freq = F, main = "Repartition de A(X)", xlab = "A(X)", ylim=c(0,3))
lines(density(x), col = "red")
legend("topright", "Densité de X", col = "red", lwd = 1, box.lty = 1)
```





On construit l'estimateur  $\hat{\delta}_n$  par la méthode antithétique :

$$\hat{\delta}_n = \frac{1}{n} \sum_{i=1}^n \frac{h(X_i) + h(A(X_i))}{2} \quad \text{où } (X_i)_{i \in [1, n]} \text{ est une suite de v.a i.i.d. suivant la loi de } X.$$

On a donc

$$\text{Var}[\hat{\delta}_n] = \frac{1}{2n} \text{Var}[h(X)](1 + \rho) = \frac{1}{2} \text{Var}[\bar{\delta}_n](1 + \rho).$$

On veut maintenant utiliser la Proposition 3.5 du cours. Ici par décroissance de  $X \mapsto -X$ , on voit facilement que  $A$  est une transformation de  $\mathbb{R}^3$  décroissante en chacune de ses coordonnées.

De plus,  $h : \mathbb{R}^3 \rightarrow \mathbb{R}$  est décroissante par décroissance de  $X \mapsto e^{-X}$ .

Ainsi, d'après la Proposition 3.5 du cours, on a  $\text{Cov}(h(X), h(A(X))) \leq 0$ .

D'où,  $1 + \rho \leq 1$ . On en déduit :  $\text{Var}[\hat{\delta}_n] \leq \frac{1}{2} \text{Var}[\bar{\delta}_n]$

On s'intéresse maintenant au facteur de réduction de variance théorique, noté  $R_1$ , de  $\hat{\delta}_n$  par rapport à  $\bar{\delta}_n$ .

L'expression de  $R_1$  est:  $R_1 = R(\hat{\delta}_n, \bar{\delta}_n) = \frac{C\sigma^2}{C_1\sigma_1^2} = \frac{C}{C_1} \frac{2}{1+\rho}$  avec  $\sigma^2 = \text{Var}[\bar{\delta}_n]$ ,  $\sigma_1^2 = \text{Var}[\hat{\delta}_n]$  et  $C$  le coût de calcul de  $\bar{\delta}_n$  et  $C_1$  de  $\hat{\delta}_n$ .

```
# Calcul de l'estimateur
```

```
h.a <- h(a, n)
delta.ant <- mean(c(h.x, h.a))
delta.ant
```

```
## [1] 0.9588342
```

```

# Calcul de l'erreur quadratique moyenne
rho <- cor(h.x, h.a)

erreur.ant <- var.h * (1 + rho) / (2 * n)
erreur.ant

## [1] 3.529868e-08

# Calcul du facteur de réduction de variance théorique
library(microbenchmark)
test <- microbenchmark::microbenchmark(h(x, n), rmvnorm(n, mu, sigma), times = 1000)
print(test)

## Unit: microseconds
##          expr      min       lq      mean median        uq      max neval
##          h(x, n)  570.2   588.7  659.5355  596.3   628.80 2428.8  1000
##  rmvnorm(n, mu, sigma) 1363.8 1382.6 1530.9234 1393.1 1430.25 5819.7  1000
C_h <- mean(test$time[which(test$expr == "h(x, n)")])
C_X <- mean(test$time[which(test$expr == "rmvnorm(n, mu, sigma)")])

R1 <- 2 * (C_h + C_X) / ((C_X + 2 * C_h) * (1 + rho))
R1

## [1] 37.28105

```

On conclut que  $\hat{\delta}_n$  est 37.2810544 fois plus efficace que  $\bar{\delta}_n$ .

#### 4.(a)

On choisit ici la loi génératrice des moments de loi mutivarée en  $(-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3})$

On a  $M_X(t) = e^{\mu^T t + \frac{1}{2} t^T \Sigma t} = \mathbb{E}[e^{<t, X>}]$

Et  $h_0(X) = e^{-\frac{1}{3} \sum_{i=1}^n X_i} = e^{<t, X>} = \exp\left\{<\begin{pmatrix} -\frac{1}{3} \\ -\frac{1}{3} \\ -\frac{1}{3} \end{pmatrix}, \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}>\right\}$

D'où  $m = \mathbb{E}[e^{<t, X>}] = e^{\mu^T t + \frac{1}{2} t^T \Sigma t}$

On en déduit finalement

$$\hat{\delta}_n(b) = \frac{1}{n} \sum_{i=1}^n (h(X_i) - b(h_0(X_i) - m))$$

```

h0 <- function(x) {
  return(exp(-colMeans(x)))
}

```

```
h0.x <- h0(x)
```

```
rho.control <- cor(h0.x, h.x)
rho.control
```

```
## [1] 0.9932131
```

On a  $cor(h_0(X), h(X))$  très proche de 1 (=0.9932131).

```

t1 <- c(-1 / 3, -1 / 3, -1 / 3)
m <- exp(t(mu) %*% t1 + 1 / 2 * t(t1) %*% sigma %*% t1) [1, 1]

```

```
cov.h.h0 <- cov(h0.x, h.x)
```

#### 4.(b)

Pour simuler  $b$ , on utilise la méthode de la *burn-in period*. Ainsi on utilise les  $l$  premières estimations de la suite  $(X_n)_{n \geq 1}$  pour estimer  $b^*$  via :

$$\hat{b}_l^* = \frac{\sum_{k=1}^l (h_0(X_k) - m)(h(X_k) - \bar{h}_l)}{\sum_{k=1}^l (h_0(X_k) - m)^2}$$

On utilise ensuite les  $n - l$  estimations restantes pour calculer  $\hat{\delta}_{n-l}(\hat{b}_l^*)$ . D'après le cours, nous savons que  $l$  doit rester relativement petit face à  $n - l$ .

On fait varier donc  $l$  entre 0 et 500, pour trouver la valeur de  $l$  qui nous donne l'estimateur avec la plus faible erreur quadratique.

En traçant l'évolution de  $b$  et de l'erreur en fonction de  $l$ , on voit que l'erreur n'évolue pas trop et que  $b$  stagne vite autour de la valeur 1.

```
essai <- seq(from = 0, to = 500, by = 1)
err <- c()
del <- c()
b <- c()
for (l in essai) {
  h.x.l <- h.x[1:l]
  h.x.nl <- h.x[l:n]

  h0.x.l <- h0.x[1:l]
  h0.x.nl <- h0.x[l:n]
  bl <- sum((h0.x.l - m) * (h.x.l - mean(h.x.l))) / sum((h0.x.l - m) * (h0.x.l - m))

  d <- mean(h.x.nl - bl * (h0.x.nl - m))

  e <- var(h.x.nl) / (n - 1) + (bl * bl * var(h0.x) - 2 * bl * cov(h.x.nl, h0.x.nl)) / (n - 1)

  err <- append(err, e)
  b <- append(b, bl)
  del <- append(del, d)
}
```

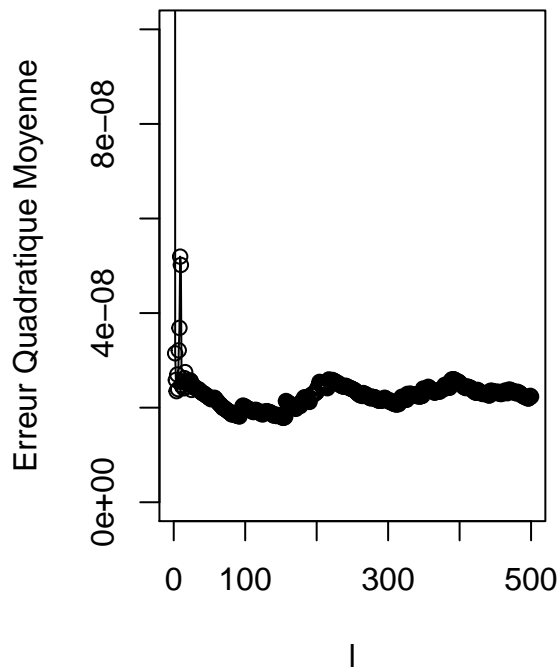
```
par(mfrow = c(1, 2))

plot(essai, err, xlab = "l", ylab = "Erreur Quadratique Moyenne",
     main = "Evolution de l'erreur en fonction de l", ylim = c(0, 1e-07))

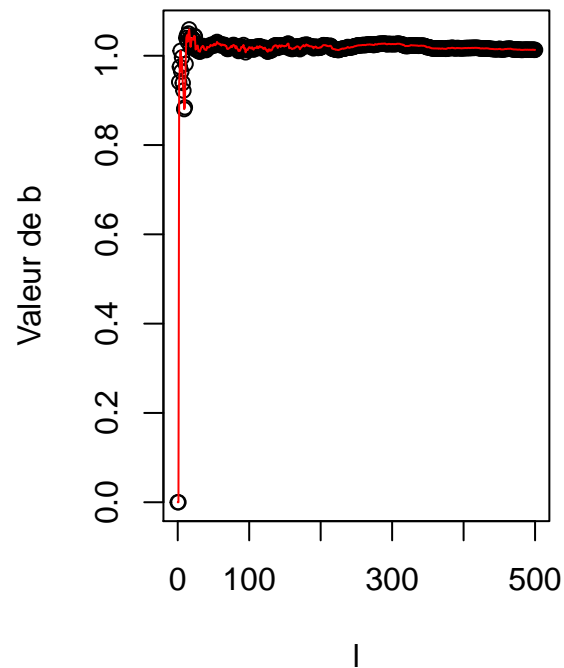
lines(essai, err)

plot(essai, b, xlab = "l", ylab = "Valeur de b", main = "Evolution de b en fonction de l")
lines(essai, b, col = "red")
```

## Evolution de l'erreur en fonction de l



## Evolution de b en fonction de l



```
w <- which(err == min(err))
essai[w] #Valeur de l
```

```
## [1] 153
```

```
err[w] #Erreur minimale
```

```
## [1] 1.783843e-08
```

```
b[w] #Valeur de b
```

```
## [1] 1.025162
```

```
del[w] #Valeur de l'estimateur
```

```
## [1] 0.9586987
```

Dans ce cas précis, nous aurions donc choisi,  $l = 153$  qui nous donne  $\hat{\delta}_{n-l}(\hat{b}_l^*) = 0.9586987$  et  $\hat{b}_l^* = 1.0251624$ .

La méthode de contrôle permet encore de diminuer notre variance et est donc encore plus performante.

## Solution Exercice 3.

### Travail préliminaire

On remarque très vite que la loi géométrique liée aux fonctions du langage R *pgeom()*, *dgeom()*, *qgeom()* et *rgeom()* avec  $\text{prob} = p$  suit la densité :

$$p(x) = p(1 - p)^x$$

pour  $x = 0, 1, 2, \dots$

Alors que la loi géométrique classique évoquée dans l'énoncé suit la densité :

$$p(x) = p(1 - p)^{x-1}$$

pour  $x = 1, 2, 3, \dots$

On introduit donc des fonctions suivant cette loi géométrique pour cet exercice :

```
rgeom_modif <- function(n, p = 0.2) {
  return(rgeom(n, p) + 1)
}

dgeom_modif <- function(n, p = 0.2) {
  return(dgeom(n - 1, p))
}

pgeom_modif <- function(n, p = 0.2) {
  return(pgeom(n - 1, p))
}
```

### 1.

L'estimateur de Monte Carlo classique est donné par :

$$\bar{\delta}_n = \frac{1}{n} \sum_{k=1}^n \sum_{i=1}^{Y_k} \log(1 + X_i)$$

avec  $(X_n)_{n \geq 1}$  i.i.d suivant une loi  $\Gamma(m, \theta)$  et  $(Y_k)_{k \in [1, n]}$  i.i.d suivant une loi  $\mathcal{G}(p)$ .

```
n <- 10000
```

```
m <- 2
```

```
p <- 0.2
```

```
theta <- 2
```

```
estim_MC <- function(n, p = 0.2) {
  ans <- c()
  y <- rgeom_modif(n, p)
  for (i in 1:n) {
    ans <- append(ans, sum(log(1 + rgamma(y[i], m, theta))))
  }
  return(list("delta" = mean(ans), "Var" = var(ans) / n, "Erreur" = var(ans) / n^2))
}
```

```
MC <- estim_MC(n)
```

```
MC
```

```
## $delta
## [1] 3.101019
##
## $Var
## [1] 0.0008276195
##
## $Erreur
## [1] 8.276195e-08
```

On a  $\bar{\delta}_n = 3.1010193$  avec une erreur quadratique moyenne valant  $8.2761947 \times 10^{-8}$ .

## 2.(a)

On cherche à estimer  $\delta = \mathbb{E}[S]$  avec  $S = \sum_{i=1}^Y \log(X_i + 1)$  avec  $(X_n)_{n \geq 1}$  i.i.d suivant une loi  $\Gamma(m, \theta)$

Dans cette question, on souhaite estimer  $\delta$  à l'aide de la méthode de stratification avec  $L = 15$  strates bien choisies.

### Choix des strates

Il paraît évident de choisir  $Y$  comme variable de stratification. En effet, comme  $Y$  suit une loi  $\mathcal{G}(p)$ ,  $Y$  est à valeur dans  $\mathbb{N}^*$ . Il faut donc créer 15 ensembles  $D_k$  formant une partition de  $\mathbb{N}^*$ .

Donc cela revient donc à estimer

$$\mathbb{E}[S] = \sum_{k=1}^L \mathbb{P}(Y \in D_k) \mathbb{E}[S|Y \in D_k]$$

Pour  $1 \leq k \leq 14$ , on choisit  $D_k = \{k\}$ . On a bien  $\mathbb{P}(Y \in D_k) = \mathbb{P}(Y = k) > 0$ .

Pour  $k = 15$ , on prend donc toutes les autres valeurs de  $\mathbb{N}^*$  donc  $\mathbb{N}^* \setminus \cup_{k=1}^{14} D_k = \{15, 16, \dots\}$ . On a bien  $\mathbb{P}(Y \in D_{15}) = \mathbb{P}(Y \geq 15) > 0$ .

Ainsi,

$$\mathbb{E}[S] = \sum_{k=1}^{14} \mathbb{P}(Y = k) \mathbb{E}[S|Y = k] + \mathbb{P}(Y \geq 15) \mathbb{E}[S|Y \geq 15]$$

### Allocation proportionnelle

Nous voulons utiliser la méthode de stratification avec allocation proportionnelle, nous allons créer une suite de  $(n_1, \dots, n_L)$  représentant les cardinaux des  $D_k$ .

Pour  $1 \leq k \leq 14$ , on prend  $n_k = n \mathbb{P}(Y = k) = np(1-p)^{k-1}$  avec  $p = 0.2$ .

Pour  $k = 15$ ,  $n_{15} = n - \sum_{k=1}^{14} n_k$

### Simulation selon $\mathcal{L}(S|Y \in \mathbf{D}_k)$

Nous voulons à présent simuler selon la loi  $\mathcal{L}(S|Y \in \mathbf{D}_k)$  pour  $1 \leq k \leq 15$ .

Une nouvelle fois pour  $1 \leq k \leq 14$  cela est plutôt facile, il suffit de simuler de construire  $n_k$  fois :

$$S_i^{(k)} = \sum_{j=1}^{n_k} \log(1 + X_j) \forall i \in [1, n_k] \quad \text{avec } (X_n)_{n \geq 1} \stackrel{iid}{\sim} \Gamma(m, \theta)$$

Pour  $k = 15$ , la tâche est plus compliquée on utilise alors la formule du cours :

Soit  $X$  une variable aléatoire réelle de fonction de répartition  $F$ . Pour  $k = 1, \dots, L$ ,  $D_k = ]d_k, d_{k+1}]$  avec  $d_k$  des constantes réelles. Si  $U \sim \mathbf{U}([0, 1])$ , alors  $X_{(k)} = F^{\leftarrow}[F(d_k) + U(F(d_{k+1}) - F(d_k))]$ ,  $k = 1, \dots, K$  suit la loi de  $X | X \in D_k$ .

Dans notre cas,  $S_i^{(15)} = F^{\leftarrow}[F(15) + U(1 - F(15))]$  suit la loi de  $\mathcal{L}(S|Y \in \mathbf{D}_{15})$ .

Enfin nous avons construit notre estimateur

$$\hat{\delta}_n(n_1, \dots, n_L) = \sum_{i=1}^L \mathbb{P}(Y \in \mathbf{D}_k) \frac{1}{n_k} \sum_{j=1}^{n_k} S_j^{(k)}$$

## 2.(b)

```
estim_Strat <- function(n,L=15,p=0.2) {
  L <- 15
  # Allocation proportionnelle
  nk <- c()
  pk <- dgeom_modif(1:(L - 1), p)
  nk <- floor(n * pk)
  pk <- append(pk, 1 - pgeom_modif(L, p))
  nk <- append(nk, n - sum(nk))
  estim <- c()
  erreur <- c()
  for (i in 1:(L - 1)) { # Simulation des Si(k)
    sk <- replicate(nk[i], sum(log(1 + rgamma(i, theta, m))))
    estim <- append(estim, pk[i] * mean(sk))
    erreur <- append(erreur, pk[i] * var(sk))
  }
  U <- runif(nk[L])
  P_Y15 <- pgeom_modif(L, p)
  # Simulation de n15 Y suivant la loi (Y >= 15)
  Y_cond <- qgeom(P_Y15 + (1 - P_Y15) * U, p)

  sL <- c()
  # Simulation de n15 Si(15)
  for (i in Y_cond) {
    sk <- sum(log(1 + rgamma(i, theta, m)))
    sL <- append(sL, sk)
  }
  estim <- append(estim, pk[L] * mean(sL))
  erreur <- append(erreur, pk[L] * var(sL))
  sum(estim)
  sum(erreur) / n
  return(list("delta" = sum(estim), "Var" = sum(erreur) / n, "Erreur" = sum(erreur) / n^2))
}

Strat <- estim_Strat(n)
Strat

## $delta
## [1] 3.08215
##
## $Var
## [1] 7.655507e-05
##
## $Erreur
## [1] 7.655507e-09
```

Ici  $\hat{\delta}_n(n_1, \dots, n_L) = 3.0821503$ . Et l'erreur quadratique moyenne est de  $7.6555072 \times 10^{-9}$ .

### Calcul du facteur de réduction de variance

```
library(microbenchmark)
test <- microbenchmark(estim_MC(n), estim_Strat(n))
print(test)

## Unit: milliseconds
##      expr      min       lq      mean   median      uq      max neval
## estim_MC(n) 135.4415 141.68200 148.37015 145.6270 149.58870 186.4807   100
## estim_Strat(n) 42.4854 45.98175 48.81409 47.6426 49.23905 72.9176   100

C_MC <- mean(test$time[which(test$expr == "estim_MC(n)")])
C_Strat <- mean(test$time[which(test$expr == "estim_Strat(n)")])

R <- (MC$Var * C_MC) / (C_Strat * Strat$Var)
R

## [1] 32.85928
```

L'estimateur  $\hat{\delta}_n(n_1, \dots, n_L)$  est 32.8592829 plus fois efficace que l'estimateur de Monte Carlo classique.