

TP2 Applications de la simulation probabiliste

DIDIER Thomas & GAUDIN Emma

L'objectif de ce TP est d'étudier, à travers la simulation, les notions de convergence des variables aléatoires (en probabilité et presque sûre) et les techniques d'intégration numérique basées sur la méthode de Monte Carlo.

La première partie se concentre sur l'illustration, par des simulations, des concepts de convergence en probabilité et presque sûre.

Nous abordons ensuite l'intégration par des méthodes de Monte Carlo pour estimer des intégrales et déterminer les intervalles de confiance associés en s'appuyant sur le TCL.

```
library(ConvergenceConcepts)

## Warning: le package 'ConvergenceConcepts' a été compilé avec la version R 4.5.2

## Le chargement a nécessité le package : tcltk

## Le chargement a nécessité le package : tkrplot

## Warning: le package 'tkrplot' a été compilé avec la version R 4.5.2

## Le chargement a nécessité le package : lattice

library(plotrix)

## Warning: le package 'plotrix' a été compilé avec la version R 4.5.2
```

1- Convergence de v.a. et simulation probabiliste

1.1 Rappels : loi faible et forte des grands nombres

1.2 Illustrations des convergences en probabilité et p.s.

Application 1 (Estimer la valeur d'une probabilité par une fréquence)

Soit X une v.a. réelle et E un borélien quelconque de \mathbb{R} . Si $h(\cdot) = \mathbf{1}_E(\cdot)$, alors $h(X)$ suit une loi de Bernoulli de paramètre $p := \mathbb{P}\{h(X) = 1\} = \mathbb{P}\{X \in E\}$. Dans ce cas, la Loi Faible des Grands Nombres (LFGN) nous dit que l'on peut estimer cette probabilité par la fréquence d'observation de valeurs dans E pour un n -échantillon X_1, \dots, X_n de la loi de X :

$$\overline{\mathbf{1}_E(X)}_n = \frac{1}{n} \sum_{k=1}^n \mathbf{1}_E(X_k(\omega))$$

ou encore par la proportion de succès sur une réalisation $\mathbf{1}_E(x_1), \dots, \mathbf{1}_E(x_n)$ de $\mathbf{1}_E(X_1), \dots, \mathbf{1}_E(X_n)$.

Dans cette section nous illustrerons la convergence en probabilité.

Pour cela nous considérons un échantillon $(X_n)_{n \geq 1}$ d'une loi uniforme sur $[1,9]$. Pour tout $\hat{b}n := 2\overline{X}_n - 1$

```
# code R1
gen_bn = function(n){
  X=runif(n,1,9) #Génération de l'échantillon i.i.d. Les Xk sont générées selon une loi uniforme sur [1, 9]
  Mn=cumsum(X)/(1:n) #calcule la moyenne empirique Mn = Sn / n (avec Sn = X1 + ... + Xn).
  b=2*Mn-1 # bn := 2Mn-1.
  return(b-9)
}
#Utilisation de ConceptsConvergence
check.convergence(50000,500,gen_bn,mode="p")

## $ID
## [1] ".1"
##
## $env
## <environment: 0x0000017d2e8920b0>
##
## attr(,"class")
## [1] "tkwin"
```

1. Dans le code précédent l'appel à la fonction check.convergence (n=50 000 et M = 500) permet l'obtention d'un beau graphique sur lequel on peut remarquer que pour un n très grand $\hat{p}n$ et $\hat{a}n$ tendent vers 0. On peut donc en déduire qu'il semble que $(\hat{b}n - 9) \xrightarrow[\text{proba}]{} 0$ (grâce a $\hat{p}n$) et $(\hat{b}n - 9) \xrightarrow[\text{p.s.}]{} 0$ (grâce a $\hat{a}n$). Ce qui revient à

$$\begin{aligned}(\hat{b}n) &\overset{\text{proba}}{\longrightarrow} 9 \\(\hat{b}n) &\overset{\text{Pp.s.}}{\longrightarrow} 9\end{aligned}$$

2. Montrons que $(\hat{b}n) \overset{\text{proba}}{\longrightarrow} 9$ par la théorie:

D'après la loi faible des grands nombres on a:

Soit $(Xn)_{n \geq 1}$ une suite de v.a.r.i.i.d. admettant un moment d'ordre 1 avec $\mathbb{E}[X_1]=m$ on a alors $\overline{X}_n \overset{\text{proba}}{\longrightarrow} m$.

Ici les $(Xn)_{n \geq 1}$ suivent une loi uniforme sur [1,9]. On a donc

$$\begin{aligned}\mathbb{E}[X_1] &= \int_{-\infty}^{\infty} x f_{X_1}(x) \, dx \\&= \int_1^9 x \cdot \frac{1}{9-1} \, dx \\&= \frac{1}{8} \left[\frac{x^2}{2} \right]_1^9 \\&= 5\end{aligned}$$

Donc

$$\overline{X}_n \overset{\text{proba}}{\longrightarrow} 5$$

Soit h :

$$\begin{aligned}\mathbb{R} &\rightarrow \mathbb{R} \\x &\mapsto 2x - 1\end{aligned}$$

, h est continue.

Si $X_n \overset{\text{proba}}{\longrightarrow} X$ alors $h(X_n) \overset{\text{proba}}{\longrightarrow} h(X)$ par les propositions du cours.

Donc

$$\hat{b}n := 2\overline{X}_n - 1 \overset{\text{proba}}{\longrightarrow} 2 * 5 - 1 = 9$$

Ensuite montrons que nous avons aussi la convergence $\mathbb{P}p. s. .$

Cette fois ci d'après la loi forte des grand nombre avec les mêmes arguments: Soit $(Xn)_{n \geq 1}$ une suite de v.a.r.i.i.d. admettant un moment d'ordre 1 avec $\mathbb{E}[X_1]=m$ on a alors $\overline{X}_n \overset{\text{Pp.s.}}{\longrightarrow} m$. Ici les $(Xn)_{n \geq 1}$ suivent une loi uniforme sur [1,9].

On a donc

$$\overline{X}_n \overset{\text{Pp.s.}}{\longrightarrow} 5$$

Soit h :

$$\begin{aligned}\mathbb{R} &\rightarrow \mathbb{R} \\x &\mapsto 2x - 1\end{aligned}$$

,h est continue.

$$\hat{b}n := 2\overline{X}_n - 1 \overset{\text{Pp.s.}}{\longrightarrow} 2 * 5 - 1 = 9$$

Nos résultats théoriques sont bien en accord avec les résultats conjecturés grâce à la méthode check.convergence

Dans cette section nous illustrerons la convergence Pp.s.

Pour cela nous condérons un échantillon $(Xk)_{k \geq 1}$ d'une loi uniforme sur [0,1].

Pour tout $n \geq 1$ on pose $M_n = max_{k=1,...,n} X_k$.

On souhaite explorer la convergence p.s. de $(M_n)_{k \geq 1}$

Pour cela nous créons une fonction gen_Mn qui génère pour n donné, les valeurs de M_1, M_2, \dots, M_n

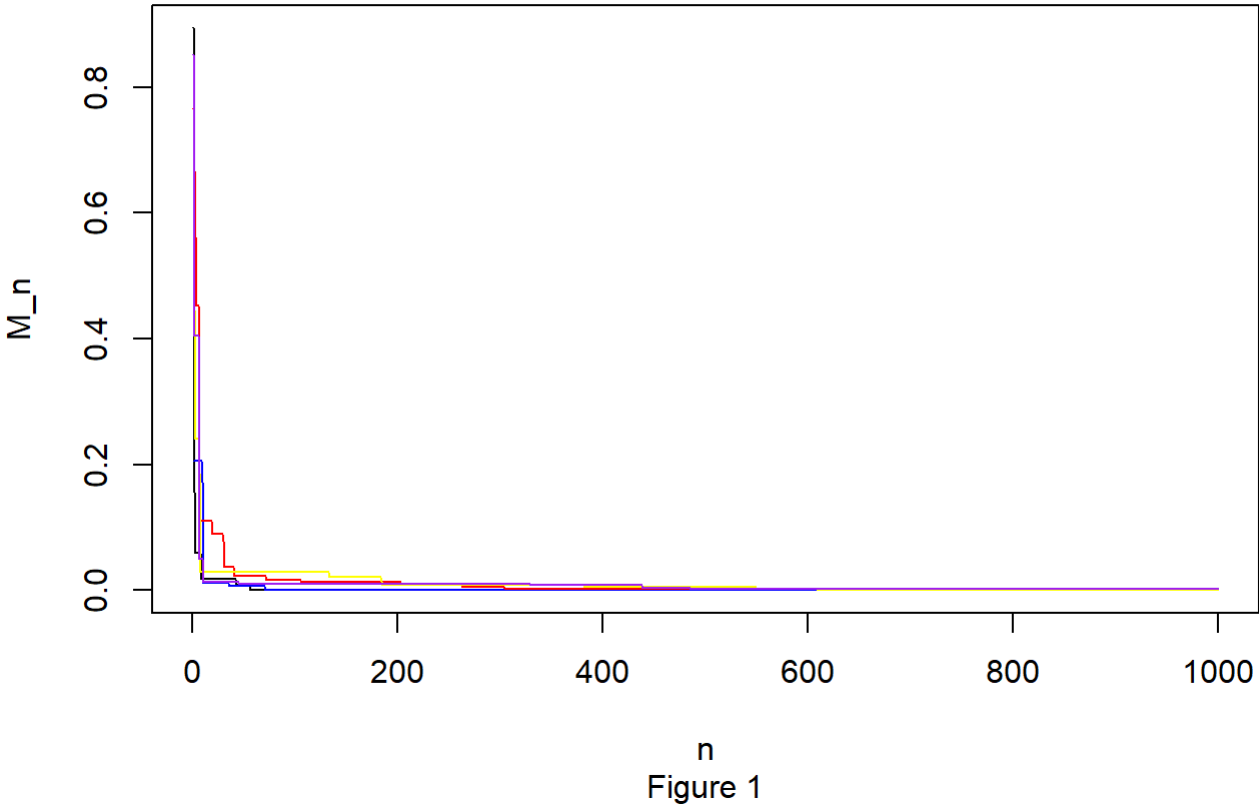
```
#code R2

#gen_Mn qui g n re pour n donn , Les valeurs de M1,M2,...,Mn
gen_Mn=function(n){
  res=cummax(runif(n))#prend Le max de n nombres al atoires entre [0,1]
  return (abs(res-1))# L' tude de |M_k - 1| vers   est  quivalente   l' tude de la convergence p.s.
  # de M_k vers 1, que l'on souhaite illustrer.
}

n=1000
plot (1:n,gen_Mn(n), type="l",xlab = "n", ylab = "M_n",
      main = "Plusieurs r alisations de M_n",
      sub = "Figure 1")#on trace la graphique de notre fonction gen_Mn en fonction des n

rep= replicate(4, gen_Mn(n))# on cr   5 r pliques ind pendantes
# On ajoute Les lignes de chaque colonne de la matrice
color <- c("blue", "red", "yellow", "purple")
for (i in 1:4) {
  lines(1:n, rep[, i], col = color[i]) #on ajoute nos nouvelles r pliques au graphique
}
```

Plusieurs r alisations de M_n



Sur le graphique g n r  par le code ci dessus (Figure 1), nous voyons que les 5 repliques de Mn tendent vers 0 quand n est grand. Nous pouvons donc conjecturer que $|M_n - 1|$ tend Pp.s. vers 0, soit que Mn tend Pp.s. vers 1. Ce r sultat est assez logique puisque la suite $Mn_{1 \leq n \leq 100}$ repr sente la valeur maximale des n tirages d'une loi uniforme sur [0,1].

Nous comparons ensuite notre r sultat avec le r sultat obtenu avec la fonction check.convergence.

```
#code R2
#Utilisation de ConceptsConvergence
check.convergence(1000,100,gen_Mn)
```

```
## $ID
## [1] ".2"
##
## $env
## <environment: 0x0000017d38e160b0>
##
## attr(,"class")
## [1] "tkwin"
```

On voit ici sur le graphique de droite (g n rer par le package ConvergenceConcepts) que $\hat{a}_n := M_n - 1$ tend vers 0. Cela signifie que la suite $(M_n - 1)_{n \geq 1}$ converge p.s. vers 0 soit que la suite $(M_n)_{n \geq 1}$ converge $\mathbb{P} - ps$ vers 1.

En effet ce r sultat se prouve aussi th oriquement :

Soit $\{X_k\}_{k \geq 1}$  chantillon d'une loi uniforme sur $[0, 1]$ et soit

$$Z_n := \sup_{k \geq n} |M_k - 1|, \quad n \geq 1$$

o  $M_n := \max(X_1, \dots, X_n)$.

Montrons que $Z_n \xrightarrow{\text{proba}} 0$.

$$Z_n \xrightarrow{\text{proba}} 0 \iff \forall \epsilon > 0 \quad \lim_{n \rightarrow \infty} P(|M_n - 1| \geq \epsilon) = 0.$$

On commence par calculer $P(|M_n - 1| \geq \epsilon)$.

Puisque $M_n \leq 1$ par définition ($X_k \in [0, 1]$), nous avons $M_n - 1 \leq 0$. Donc, $|M_n - 1| = -(M_n - 1) = 1 - M_n$.

L'événement $|M_n - 1| \geq \epsilon$ est équivalent à $1 - M_n \geq \epsilon$, soit $M_n \leq 1 - \epsilon$.

$$P(|M_n - 1| \geq \epsilon) = P(M_n \leq 1 - \epsilon)$$

Puisque M_n est le maximum des X_k , on a:

$$\begin{aligned} P(M_n \leq 1 - \epsilon) &= P(\max(X_1, \dots, X_n) \leq 1 - \epsilon) \\ &= P(X_1 \leq 1 - \epsilon, X_2 \leq 1 - \epsilon, \dots, X_n \leq 1 - \epsilon) \end{aligned}$$

Comme les variables X_k sont indépendantes et identiquement distribuées (i.i.d.) de loi $U([0, 1])$:

$$\forall k \in \{1, \dots, n\}, \quad F_{X_k}(x) = P(X_k \leq x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x > 1 \end{cases}$$

Donc :

$$P(M_n \leq 1 - \epsilon) = \prod_{k=1}^n P(X_k \leq 1 - \epsilon) = (1 - \epsilon)^n$$

pas indépendance des variables X_k

Puisque $0 < 1 - \epsilon < 1$, on a :

$$\lim_{n \rightarrow \infty} P(|M_n - 1| \geq \epsilon) = \lim_{n \rightarrow \infty} (1 - \epsilon)^n = 0$$

Ainsi, M_n converge en probabilité vers 1: $M_n \xrightarrow{P} 1$.

Démonstration de la convergence presque sûre (p.s.) :

Soit $A_n(\epsilon) = \{|M_n - 1| \geq \epsilon\}$ est :

$$\sum_{n=1}^{\infty} P(|M_n - 1| \geq \epsilon) = \sum_{n=1}^{\infty} (1 - \epsilon)^n$$

Puisque $0 < 1 - \epsilon < 1$, cette série géométrique converge.

D'après le Lemme de Borel-Cantelli,

$$\begin{aligned} \sum_{n=1}^{\infty} P(A_n) < \infty &\Rightarrow \mathbb{P}\left(\bigcap_{K=1}^{+\infty} \bigcup_{k=K}^{+\infty} A_k\right) = 0 \quad (\text{Par Borel-Cantelli}) \\ &\Rightarrow \mathbb{P}(\limsup_{n \rightarrow \infty} \{|M_n - 1| \geq \epsilon\}) = 0 \end{aligned}$$

Donc par définition:

$$Z_n = \sup_{k \geq n} |M_k - 1| \xrightarrow{p.s.} 0 \Rightarrow Z_n = \sup_{k \geq n} |M_k - 1| \xrightarrow{\text{proba}} 0$$

Donc par la proposition 3.3.4 on a que

$$M_n \xrightarrow{p.s.} 1$$

Dans cette sectio, nous illustrerons de la différence entre convergence en probabilité et Pp.s.

Considérons la suite de v.a. indépendantes $\{X_n\}_{n \geq 1}$ où X_n suit la loi $Ber(1/\sqrt{n})$.

Intuitivement, on pourrait penser que $X_n \xrightarrow{\text{proba}} 0$ et $X_n \xrightarrow{\mathbb{P}-ps} 0$. On cherche alors à vérifier ces hypothèses en utilisant le package ConvergenceConcepts, et en traçant certaines réalisations de X_n lorsque n est grand.

```
#code R3

#fonction qui génère la v.a. Xn qui suit la Loi Ber(1/sqrt(n)) .
Xn = function(n){
  rbinom(1, size =1, prob = 1/sqrt(n))
}

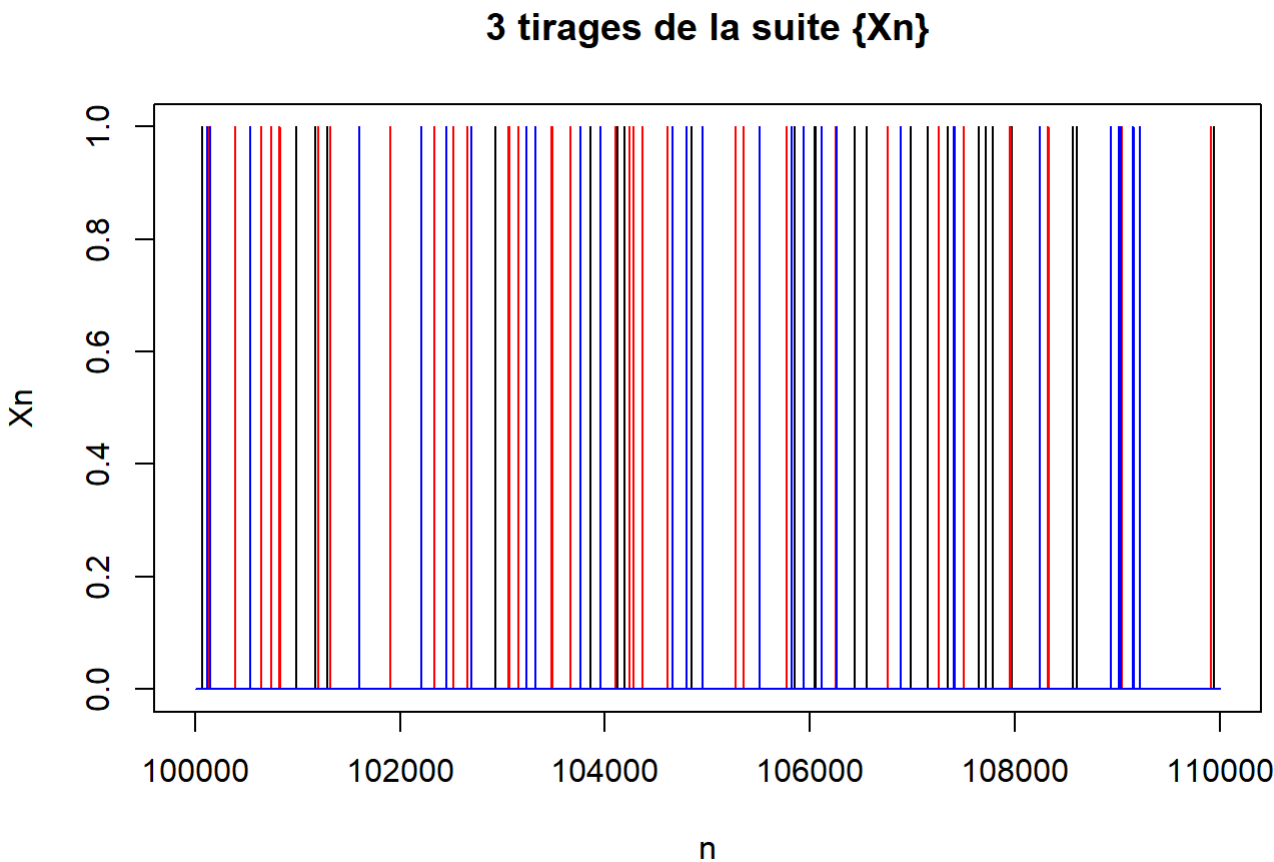
#fonction genXn qui génère un vecteur aléatoire (X(1+1e+5),X(2+1e+5),...,X(n+1e+5))
genXn = function(n){
  sapply((1+1e+5):(n+1e+5), Xn)
}

#Utilisation de ConceptsConvergence
check.convergence(nmax=1e+3,M=1e+3,genXn=genXn)
```

```
## $ID
## [1] ".3"
##
## $env
## <environment: 0x0000017d37fb39d8>
##
## attr(,"class")
## [1] "tkwin"
```

```
#plot des valeurs Xn pour 3 tirages de la suite {Xn} lorsque n est grand
n=1e+4
plot ((1+1e+5):(n+1e+5),genXn(n), type="l",
      xlab = "n", ylab = "Xn",main = "3 tirages de la suite {Xn}")

lines((1+1e+5):(n+1e+5),genXn(n),col="red")
lines((1+1e+5):(n+1e+5),genXn(n),col="blue")
```



Nous commençons le code par l'implémentation de la fonction **Xn(n)** qui génère un simple tirage suivant la loi X_n . Cette fonction nous permet d'utiliser **sapply** dans la fonction **genXn(n)** qui prend un entier en entrée et qui renvoi un vecteur aléatoire $(X_{1+1e+5}, X_{2+1e+5}, \dots, X_{n+1e+5})$. On évite alors une boucle couteuse en calcul. Ainsi on peut observer ce qu'il se passe lorsque n est grand, sans pour autant générer des vecteur de trop grande dimension.

Nous appelons ensuite la fonction **check.convergence** pour visualiser les différents type de convergence. Nous tracerons aussi un graphique (Figure 2) affichant 3 tirages de la suite X_n pour n compris entre 100000 et 101000.

On peut observer que \hat{p}_n semble proche de 0, et donc qu'il y aurait bien une convergence en probabilité. Le graphique semble beaucoup moins convaincant quant à la convergence presque sure, car bien que n soit grand, il y a toujours l'apparition d'événement $[X_n = 1]$.

Montrons maintenant ces hypothèses de manière théorique.

Convergence en probabilité :

$\forall \varepsilon > 0,$

$$\begin{aligned} \mathbb{P}(|X_n| \geq \varepsilon) &= \mathbb{P}(X_n = 1), \text{ car } X_n(\Omega) = \{0, 1\} \\ &= \frac{1}{\sqrt{n}} \xrightarrow[n \rightarrow \infty]{} 0 \end{aligned}$$

Ainsi on a $X_n \xrightarrow{\text{proba}} 0$.

Convergence presque sure :

On dit que $X_n \xrightarrow{\mathbb{P}-ps} 0$ si il existe Ω_0 un événement tel que $\mathbb{P}(\Omega_0) = 1$ et

$$\forall \omega \in \Omega_0, \quad \lim_{n \rightarrow +\infty} X_n(\omega) = 0$$

Or l'existence d'un tel événement est impossible !

En effet,

$$\sum_{k=1}^{\infty} \mathbb{P}(X_k = 1) = \sum_{k=1}^{\infty} \frac{1}{\sqrt{k}} = +\infty \quad (\text{d'après Riemann})$$

D'après le lemme de Borel Cantelli on a alors

$$\mathbb{P}\left(\bigcap_{n=1}^{\infty}\bigcup_{k=n}^{\infty}\{X_k=1\}\right)=1$$

Cela signifie que P p.s. la suite $X_n(\omega)$ prend la valeur 1 une infinité de fois.

Ainsi X_n ne converge pas p.s. vers 0

Cet exemple illustre bien qu'il existe une différence entre la convergence en probabilité et la convergence p.s. Notamment que la convergence en probabilité n'implique pas la convergence p.s.

1.3 Rappels : convergence en loi, Théorème Central Limite (TCL).

Définition 1.3 (Convergence en loi)

La suite de v.a. réelles $\{X_n\}_{n\geq 1}$ converge vers la v.a. X , si pour tout t point de continuité de la fonction de répartition F_X de X , on a

$$\lim_{n\rightarrow +\infty} F_{X_n}(t) = \lim_{n\rightarrow +\infty} \mathbb{P}\{X_n \leq t\} = \mathbb{P}\{X \leq t\} = F_X(t).$$

Une conséquence de la LFGN : la fonction de répartition empirique associée à une suite $\{Y_k\}_{k\geq 1}$ de v.a.i.i.d. de même loi de fonction de répartition F , est telle qu'il existe $A \subset \Omega$ tel que $\mathbb{P}(A) = 1$ et

$$\forall t \in \mathbb{R}, \forall \omega \in A, F_M(t, \omega) := \frac{1}{M} \sum_{k=1}^M \mathbf{1}_{\{Y_k \leq t\}}(\omega) = \frac{1}{M} \sum_{k=1}^M \mathbf{1}_{]-\infty, t]}(Y_k(\omega)) \xrightarrow{M \rightarrow +\infty} F(t).$$

En fait, cette convergence peut être renforcée en :

Théorème 1.3 (Théorème de Glivenko-Cantelli) Pour une suite $\{Y_k\}_{k\geq 1}$ de v.a.i.i.d. de loi commune de fonction de répartition F , alors il existe $A \subset \Omega$ tel que $\mathbb{P}(A) = 1$ et

$$\forall \omega \in A, \lim_{M \rightarrow +\infty} \sup_{t \in \mathbb{R}} |F_M(t, \omega) - F(t)| = 0. \quad (5)$$

Bilan:

Dans la Définition 1.3, pour n fixé, on approche la fonction de répartition de X_n (si on ne la connaît pas) par la fonction de répartition empirique d'un M -échantillon de X_n , et, pour des valeurs croissantes de n , on compare la fonction empirique du M -échantillon de X_n à celle de X . L'intérêt de cette méthode est qu'elle s'applique quel que soit le type de v.a. considéré.

Théorème 1.4 (Théorème Central Limite)

Pour une suite $\{X_n\}_{n\geq 1}$ de v.a.i.i.d. admettant un moment d'ordre deux avec $m := \mathbb{E}[X_1]$ et $\sigma^2 = \mathbb{V}(X_1)$, définissons les v.a.

$$S_n := \sum_{k=1}^n X_k, \quad Z_n := \frac{S_n - \mathbb{E}[S_n]}{\sqrt{\mathbb{V}(S_n)}} = \sqrt{n} \left(\frac{\bar{X}_n - m}{\sigma} \right).$$

Alors la suite $\{Z_n\}_{n\geq 1}$ converge en loi vers une v.a. Z de loi $N(0, 1)$, c'est à dire

$$\forall t \in \mathbb{R}, \lim_{n \rightarrow +\infty} F_{Z_n}(t) = \lim_{n \rightarrow +\infty} \mathbb{P}\{Z_n \leq t\} = F_Z(t) := \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx.$$

Dans cette section nous allons illustrer le TCL grâce à des lois du chi-deux.

Considérons une suite de v.a. indépendantes $\{X_n\}_{n\geq 1}$ avec $X_n \sim \chi_n^2$.

Montrons que $Z_n := \frac{X_n - n}{\sqrt{2n}}$, converge en loi vers $N(0, 1)$.

On prend une suite de v.a.r indépendantes $\{Y_n\}_{n\geq 1}$ de loi normale centrée réduite, tel que $X_n = Y_1^2 + \dots + Y_n^2$. Ainsi :

$$Z_n = \frac{Y_1^2 + \dots + Y_n^2 - n}{\sqrt{2n}} = \frac{n\left(\frac{Y_1^2 + \dots + Y_n^2}{n} - 1\right)}{\sqrt{2}\sqrt{n}} = \frac{\sqrt{n}(\bar{Y}_n^2 - 1)}{\sqrt{2}}$$

On remarque que si l'on montre que $\mathbb{E}[Y_n^2] = 1$ et $\mathbb{V}[Y_n^2] = 2$, alors on pourra appliquer le TCL et montrer le résultat souhaité.

Pour ce faire, nous calculons la loi de la v.a.r. $U := Y_n^2$.

Soit I un interval quelconque de \mathbb{R} :

$$\begin{aligned} \mathbb{P}(U \in I) &= \mathbb{P}(Y_n^2 \in I) \\ &= \int_{\mathbb{R}} \mathbf{1}_I(y^2) f_{Y_n}(y) \, dy && \text{où } f_{Y_n}(y) \text{ est la densité de } Y_n \sim \mathcal{N}(0, 1) \\ &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \mathbf{1}_I(y^2) e^{-y^2/2} \, dy \\ &= \frac{2}{\sqrt{2\pi}} \int_0^{+\infty} \mathbf{1}_I(y^2) e^{-y^2/2} \, dy && \text{car l'intégrande est paire} \\ &\stackrel{u=y^2}{=} \int_I \frac{1}{\sqrt{2\pi u}} e^{-u/2} \mathbf{1}_{[0, +\infty[}(u) \, du \end{aligned}$$

On reconnait ici la fonction densité de la loi $Ga(\frac{1}{2}, \frac{1}{2})$, on en déduit donc que la v.a.r U suit cette loi.

Donc $\{U_n\}_{n\geq 1}$ est uen suite de v.a.r.i.i.d et elle admet un moment d'ordre 2. U suit la loi Ga $Ga(\frac{1}{2}, \frac{1}{2})$ on connait donc son espérance: $\mathbb{E}[U] = 1$ et sa variance : $\mathbb{V}[U] = 2$.

On lui applique le TCL, et on obtient que $Z_n \xrightarrow{\text{loi}} N(0, 1)$.

Illustrons à présent ce résultat à l'aide du code suivant :

```
#code R5

#Genere la variable aléatoire Zn
Zn = function(n){
  X = rchisq(1,df=n)
  Z = (X-n)/sqrt(2*n)
  return(Z)
}

#Genere un m-echantillon suivant la loi de Zn
vecZn_ks = function(m,n){
  return(replicate(m,Zn(n)))
}

#Vérification de la convergence en loi par Le test de Kolmogorov-Smirnov
s = 0
for (i in 1:100){
  s=s+ks.test(vecZn_ks(100,1e+6),"pnorm",mean=0,sd=1)$p.value #test de K.S.
}
p_value_moy = s/100 #calcul de la moyenne des p-value
cat("moyenne de p.value = ")
```

```
## moyenne de p.value =
```

```
print(p_value_moy)
```

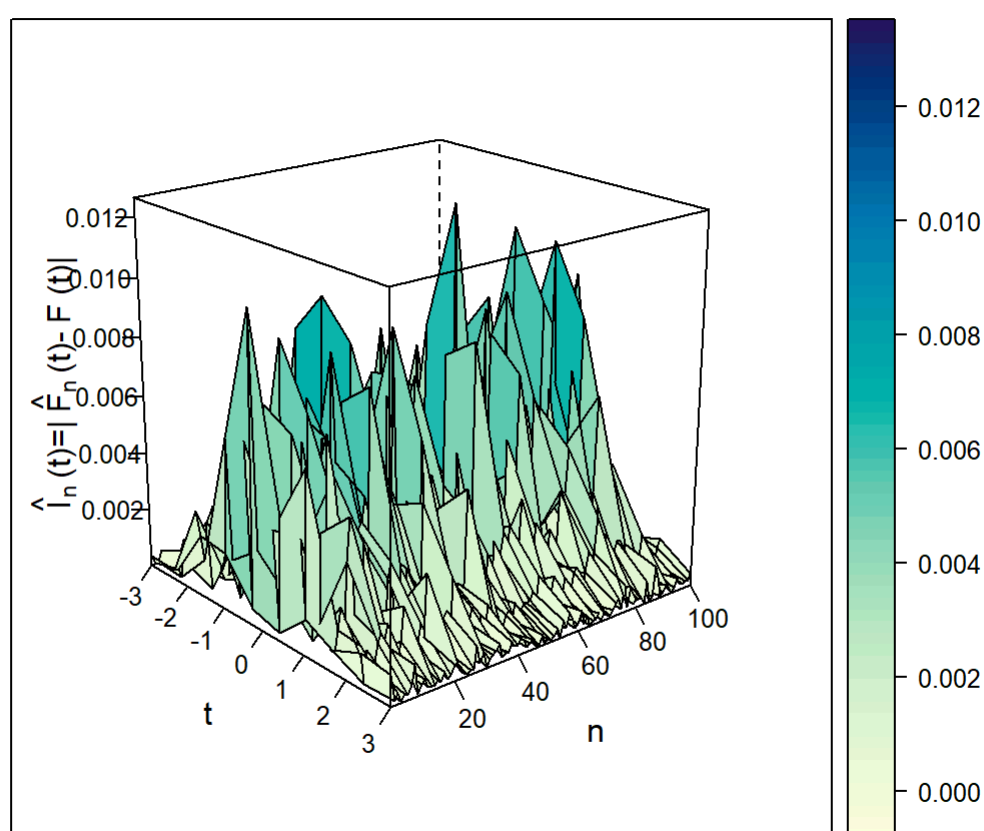
```
## [1] 0.4969336
```

```
#Illustration graphique de la convergence en loi

#Genere un vecteur aléatoire (Z(1+1e+5),...,Z(n+1e+5))
genvecZn = function(n){
  return(sapply((1+1e+5):(n+1e+5),Zn))
}

#utilisation du package ConvergenceConcepts
check.convergence(1e+2,1e+4,mode="L",
  genXn=genvecZn,
  density=FALSE,
  probfunc=function(x){pnorm(x,0,1)},
)
```

Convergence in law?



```
## $ID
## [1] ".4"
##
## $env
## <environment: 0x0000017d2dc04cf0>
##
## attr(,"class")
## [1] "tkwin"
```

Tout d’abord on implémente la fonction **Zn(n)** qui prend en paramètre un entier, et qui renvoie un unique tirage suivant la loi de Z_n . Puis l’on implémente la fonction **vecZn_ks(m,n)** qui réalise un m-échantillon suivant la loi de Z_n , afin de réaliser un test de Kolmogorov-Smirnov. On va donc réaliser 100 tests de K.S., qui vont estimer la cohérence d’un 100-échantillon suivant la loi de Z_{1e+6} avec une loi $N(0, 1)$. La valeur moyenne de p-value étant bien supérieure à 0.1, on peut conclure que les valeurs de Z_n obtenues lorsque n est grand, sont cohérentes avec une loi normale.

Pour la deuxième partie du code, on cherche à visualiser cette convergence. Pour ce faire, on implémente la fonction **genvecZn(n)** qui génère un vecteur $(Z_{1+1e+5}, Z_{2+1e+5}, \dots, Z_{n+1e+5})$, afin d’observer la loi de Z_n lorsque n est grand, sans pour autant générer un vecteur de trop grande taille. La convergence en loi parait tout autant convaincante, puisque la fonction de répartition empirique, semble superposer la fonction de répartition théorique. C’est une condition équivalente à la convergence en loi puisque la fonction de répartition de la loi normale centrée réduite est continue en tout point. On observe également cette convergence avec le plot en 3 dimensions tracé par le package ConvergenceConcepts. En effet pour des valeurs de n qui sont grandes (de l’ordre de 10^5) l’écart entre les 2 fonctions de répartitions (empirique et théorique) est faible, de l’ordre de 10^{-2} .

2- Intégration par des méthodes de Monte Carlo

La LFGN permet d’obtenir des estimations d’une espérance à partir d’une simulation probabiliste donc d’obtenir une valeur approchée d’un certain nombre d’intégrales ou de somme de série en fonction du caractère continu ou discret des v.a. concernées. En effet, il suffit que l’intégrale

$$I := \int_{\mathbb{R}} g(x) \, dx \quad (7a)$$

avec $\int_{\mathbb{R}} |g(x)| \, dx < +\infty$ soit interprétée comme une espérance sous la loi commune de l’échantillon, c’est à dire qu’elle se réécrit sous la forme

$$I = \int_{\mathbb{R}} h(x) f(x) \, dx \quad \text{où } h(x) := \frac{g(x)}{f(x)} \quad (7b)$$

où f est une densité de probabilité telle que $f(x) = 0 \Rightarrow g(x) = 0$. Noter que $\int_{\mathbb{R}} |h(x)| f(x) \, dx = \int_{\mathbb{R}} |g(x)| \, dx < +\infty$ et $I = \mathbb{E}[h(X)]$ où X est une v.a. admettant la densité f . Il suffit donc de simuler un n -échantillon de (X_1, \dots, X_n) de la loi de densité f , puis, par la LFGN (légitime car $\mathbb{E}[|h(X)|] < +\infty$), d’utiliser la moyenne empirique

$$\overline{h(X)}_n := \frac{1}{n} \sum_{k=1}^n h(X_k)$$

pour estimer la valeur de l’intégrale dans (7b). On adoptera la notation \mathbb{E}_f pour signaler une espérance calculée relativement à la loi de densité f .

Par ailleurs, pour mesurer la qualité de l’approximation, on construit un intervalle de confiance pour I sur la base du TCL (avec estimation de la variance). La construction pour une espérance est donnée en Annexe B. Bien entendu cela requiert que

$$\mathbb{E}_f[h(X)^2] = \int_{\mathbb{R}} \frac{g(x)^2}{f(x)^2} f(x) \, dx = \int_{\mathbb{R}} \frac{g(x)^2}{f(x)} \, dx < +\infty.$$

Notons que cette condition n’est pas garantie par $\int_{\mathbb{R}} |g(x)| \, dx < +\infty$ ou $\int_{\mathbb{R}} |g(x)|^2 \, dx < +\infty$. Sous la condition $\mathbb{E}_f[h(X)^2] < +\infty$, la variance de $\overline{h(X)}_n$ est

$$\mathbb{V}(\overline{h(X)}_n) = \frac{\mathbb{V}_f(h(X_1))}{n} = \frac{1}{n} \left[\int_{\mathbb{R}} h(x)^2 f(x) \, dx - I^2 \right].$$

Le TCL permet d’écrire

$$\forall t \in \mathbb{R}, \lim_{n \rightarrow +\infty} \mathbb{P} \left(\frac{\sqrt{n}(\overline{h(X)}_n - I)}{\sqrt{\mathbb{V}_f(h(X_1))}} \leq t \right) = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx$$

d’où l’intervalle de confiance de niveau 0.95 avec un échantillon de taille n

$$\text{IC}_{n,0.95}(I) = \left[\overline{h(X)}_n \pm 1.96 \sqrt{\frac{\mathbb{V}_f(h(X_1))}{n}} \right].$$

Comme la valeur de I est inconnue, la valeur de $\mathbb{V}(h(X_1))$ l’est également. On la remplace alors par son estimation classique par la variance empirique

$$S_n^2(h) = \overline{h(X)^2}_n - (\overline{h(X)})_n^2$$

ce qui donne, par le TCL avec estimation consistante de la variance, l'intervalle de confiance de I

$$\text{IC}_{n,0.95}(I) = \left[\overline{h(X)}_n \pm 1.96 \sqrt{\frac{S_n^2(h)}{n}} \right].$$

Considérons l'exemple jouet de la fonction $g(x) := [\cos(50x) + \sin(20x)]^2$ et l'intégrale

$$I = \int_0^1 g(x) dx = \int_0^1 [\cos(50x) + \sin(20x)]^2 dx$$

Cette dernière se réécrit sous la forme:

$$I := \int_{\mathbb{R}} g(x) \mathbf{1}_{[0,1]}(x) dx = \int_{\mathbb{R}} g(x) f(x) dx$$

où f est la densité d'une loi unif. sur l'intervalle $[0, 1]$. Il suffit donc de produire un n -échantillon X_1, \dots, X_n d'une telle loi et d'estimer l'intégrale par la somme

$$\overline{h(X)}_n = \frac{1}{n} \sum_{k=1}^n h(X_k).$$

Nous allons ensuite construire des intervalles de confiances associés à chaque estimation obtenue. On construit un intervalle de confiance pour I sur la base du TCL (avec estimation de la variance).

$$\begin{aligned} \mathbb{E}_f[h(X)^2] &= \int_{\mathbb{R}} \frac{g(x)^2}{f(x)^2} f(x) dx \\ &= \int_{\mathbb{R}} \frac{g(x)^2}{f(x)} dx \\ &= \int_0^1 \frac{g(x)^2}{1} dx \quad (\text{Car } f(x) = \mathbf{1}_{[0,1]}(x)) \\ &= \int_0^1 g(x)^2 dx \end{aligned}$$

Or la fonction $g(x)^2$ est continue sur l'intervalle fermé et borné $[0, 1]$ (car somme de deux fonction continues).

Donc l'intégrale $\int_0^1 g(x)^2 dx$ est finie.

Donc $\mathbb{E}_f[h(X)^2] < \infty$. Donc $h(X)$ admet un moment d'ordre 1, et donc $I < \infty$.

Sous la condition $\mathbb{E}_f[h(X)^2] < +\infty$, la variance de $\overline{h(X)}_n$ est

$$\mathbb{V}(\overline{h(X)}_n) = \frac{\mathbb{V}_f(h(X_1))}{n} = \frac{1}{n} \left[\int_{\mathbb{R}} h(x)^2 f(x) dx - I^2 \right].$$

Le TCL permet d'écrire

$$\forall t \in \mathbb{R}, \lim_{n \rightarrow +\infty} \mathbb{P} \left\{ \frac{\sqrt{n}(\overline{h(X)}_n - I)}{\sqrt{\mathbb{V}(h(X_1))}} \leq t \right\} = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) dx$$

d'où l'intervalle de confiance de niveau 0.95 avec un échantillon de taille n

$$\text{IC}_{n,0.95}(I) = \left[\overline{h(X)}_n \pm 1.96 \sqrt{\frac{\mathbb{V}_f(h(X_1))}{n}} \right].$$

Comme la valeur de I est inconnue, la valeur de $\mathbb{V}(h(X_1))$ l'est également. On la remplace alors par son estimation classique par la variance empirique

$$S_n^2(h) = \overline{h(X)^2}_n - (\overline{h(X)})_n^2$$

ce qui donne, par le TCL avec estimation consistante de la variance, l'intervalle de confiance de I

$$\text{IC}_{n,0.95}(I) = \left[\overline{h(X)}_n \pm 1.96 \sqrt{\frac{S_n^2(h)}{n}} \right].$$

Après avoir démontré, à l'aide de l'exemple jouet (Code R6), l'efficacité de la méthode de Monte Carlo et la validité des intervalles de confiance obtenus grâce au TCL pour une fonction à support borné (l'intervalle $[0, 1]$) où la condition $\mathbb{E}_f[h(X)^2] < +\infty$ est facilement vérifiée, nous passons maintenant à l'étude d'une intégrale I sur un support non borné (\mathbb{R}). L'objectif du Code R7 est de tester l'estimation de cette intégrale $I := \int_{\mathbb{R}} \frac{1}{1+|x|^3} dx$ en utilisant deux densités différentes (Loi de Cauchy et Loi de Laplace), afin d'illustrer l'importance critique de la condition de variance finie $\mathbb{E}_f[h(X)^2] < +\infty$ pour l'application rigoureuse du TCL et la validité des intervalles de confiance.

On souhaite dans cette section calculer l'intégrale

$$I := \int_{\mathbb{R}} \frac{1}{1 + |x|^3} \, dx.$$

Puis nous proposerons une estimation de I en introduisant:

- 1.une densité de la loi de Cauchy : $f(x) := \frac{1}{\pi(1+x^2)}$
- 2.une densité de type Laplace (voir Code R5 du TP N° 1) : $f(x) := \frac{1}{2}e^{-|x|}$.

On cherche aussi à étudier la possibilité de construire un intervalle de confiance pour ces deux cas de figure.

Commencons avec la loi de Cauchy.

Tout d'abord, vérifions que $f(x) = 0 \Rightarrow g(x) = 0$ pour que $h(x) := g(x)/f(x)$ soit bien défini lors de l'intégration par la méthode de Monte Carlo.

Cette condition est rapidement satisfaite car la densité de Cauchy $f(x) = \frac{1}{\pi(1+x^2)}$ est strictement positive sur \mathbb{R} .

Afin de garantir un intervalle de confiance non erroné, nous devons vérifier la condition essentielle pour le TCL, à savoir que l'espérance du carré de $h(X)$ est finie : $\mathbb{E}_f[h(X)^2] < \infty$.

Nous obtenons :

$$\mathbb{E}_f[h(X)^2] = \int_{\mathbb{R}} \frac{\left(\frac{1}{1+|x|^3}\right)^2}{\frac{1}{\pi(1+x^2)}} \, dx = \int_{\mathbb{R}} \frac{\pi(1+x^2)}{(1+|x|^3)^2} \, dx$$

Pour étudier la convergence, nous examinons le comportement de l'intégrande à l'infini :

$$\frac{\pi(1+x^2)}{(1+|x|^3)^2} \underset{|x| \rightarrow \infty}{\sim} \frac{\pi x^2}{(|x|^3)^2} = \frac{\pi x^2}{x^6} = \frac{\pi}{x^4}$$

Or $\int_{\mathbb{R}} \frac{1}{x^4} \, dx$ est convergente d'après Riemann. Donc l'intégrale $\mathbb{E}_f[h(X)^2]$ est également convergente. Nous pouvons donc valider l'utilisation du Théorème Central Limite pour construire l'intervalle de confiance.

```
#code R7
g=function(x){
  return (1/(1+abs(x)^3))
}

h=function(x){
  return(g(x)/dcauchy(x))
}

IC=function(n){
  X=rcauchy(n)# Génération des échantillons (X) selon La Loi q(x) = Cauchy
  I=cumsum(h(X))/(1:n) #estimation de L'espérance E(h(X))
  Var=pmax(0,(cumsum(h(X)^2)/(1:n) -I^2)) #estimation de La variance Var(h(X))
  Err=1.96*sqrt(Var/(1:n)) # Calcul de L'Erreur
  return(list(Moy=I, Err=Err))
}

n=1e4
indices=1:n
indices_CI=seq(1, n, length.out = 100)# Définition des indices pour Le tracé des IC

Résultat_IC=IC(n)

Moyenne=Résultat_IC$Moy[indices]
Error=Résultat_IC$Err[indices]

val_exacte = integrate(f = g, lower = -Inf, upper = Inf);# Calcul de La valeur exacte

cat("Valeur approchée = ")
```

Valeur approchée =

print(Moyenne[length(Moyenne)]) # approximation finale

[1] 2.412238

cat("\nValeur exacte = ")

##
Valeur exacte =

```
print(val_exacte) #valeur exacte
```

```
## 2.418399 with absolute error < 9.7e-07
```

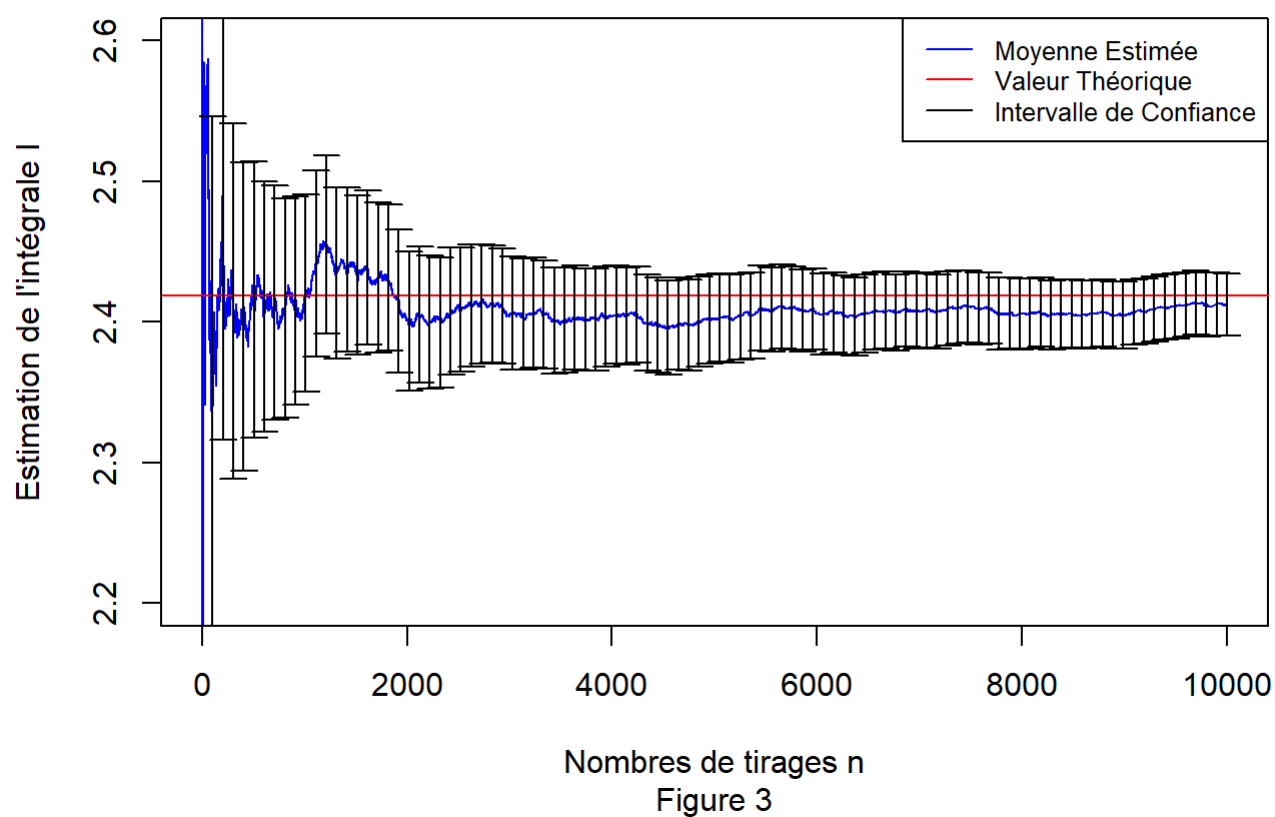
```
#Tracé Convergence de La Moyenne Estimée
plot(indices, Moyenne, type='l', col="blue",
      xlab = "Nombres de tirages n", ylab="Estimation de l'intégrale I",
      ylim = c(2.2,2.6),
      main="Convergence de I pour la loi de Cauchy",
      sub = "Figure 3")

#Tracé de La valeur exacte
abline (h=val_exacte$value, col="red")

#Tracé des Intervalles de Confiance (IC)
plotCI(indices_CI, Résultat_IC$Moy[indices_CI],
       uiw=Résultat_IC$Err[indices_CI],liw=Résultat_IC$Err[indices_CI],
       add=TRUE,lwd=1,pch=NA)

#Legende
legend("topright", legend = c("Moyenne Estimée", "Valeur Théorique", "Intervalle de Confiance"),
      col = c("blue", "red", "black"),
      lty = c(1, 1, 1),lwd = c(1, 1, 1), cex = 0.8)
```

Convergence de I pour la loi de Cauchy



Sur le graphique de la Figure 3, on retrouve bien que plus n est grand, plus l'intervalle de confiance diminue et l'estimation devient donc de plus en plus précise. Cette estimation converge vers une valeur de I représentée en rouge. On peut donc estimer la valeur de I à 2.418399 (calculée grâce à la fonction `integrate`). Observons maintenant lorsque n est plus grand, pour cela on se muni du code suivant.

```
#code R7
#même chose que le programme précédent mais avec n plus grand, on fait un "zoom"

g=function(x){
  return (1/(1+abs(x)^3))
}

h=function(x){
  return(g(x)/dcauchy(x))
}

IC=function(n){
  X=rcauchy(n)# Génération des échantillons (X) selon la loi  $q(x) = \text{Cauchy}$ 
  I=cumsum(h(X))/(1:n) #estimation de l'espérance  $E(h(X))$ 
  Var=pmax(0,(cumsum(h(X)^2)/(1:n) -I^2)) #estimation de la variance  $\text{Var}(h(X))$ 
  Err=1.96*sqrt(Var/(1:n)) # Calcul de l'Erreur
  return(list(Moy=I, Err=Err))
}
n=1e6
indices=1:n
indices_CI=seq(1, n, length.out = 100)# Définition des indices pour le tracé des IC

Résultat_IC=IC(n)

Moyenne=Résultat_IC$Moy[indices]
Error=Résultat_IC$Err[indices]

val_exacte = integrate(f = g, lower = -Inf, upper = Inf);# Calcul de la valeur exacte

cat("Valeur approchée = ")
```

```
## Valeur approchée =
```

```
print(Moyenne[length(Moyenne)]) # approximation finale
```

```
## [1] 2.418554
```

```
cat("\nValeur exacte = ")
```

```
##
## Valeur exacte =
```

```
print(val_exacte) #valeur exacte
```

```
## 2.418399 with absolute error < 9.7e-07
```

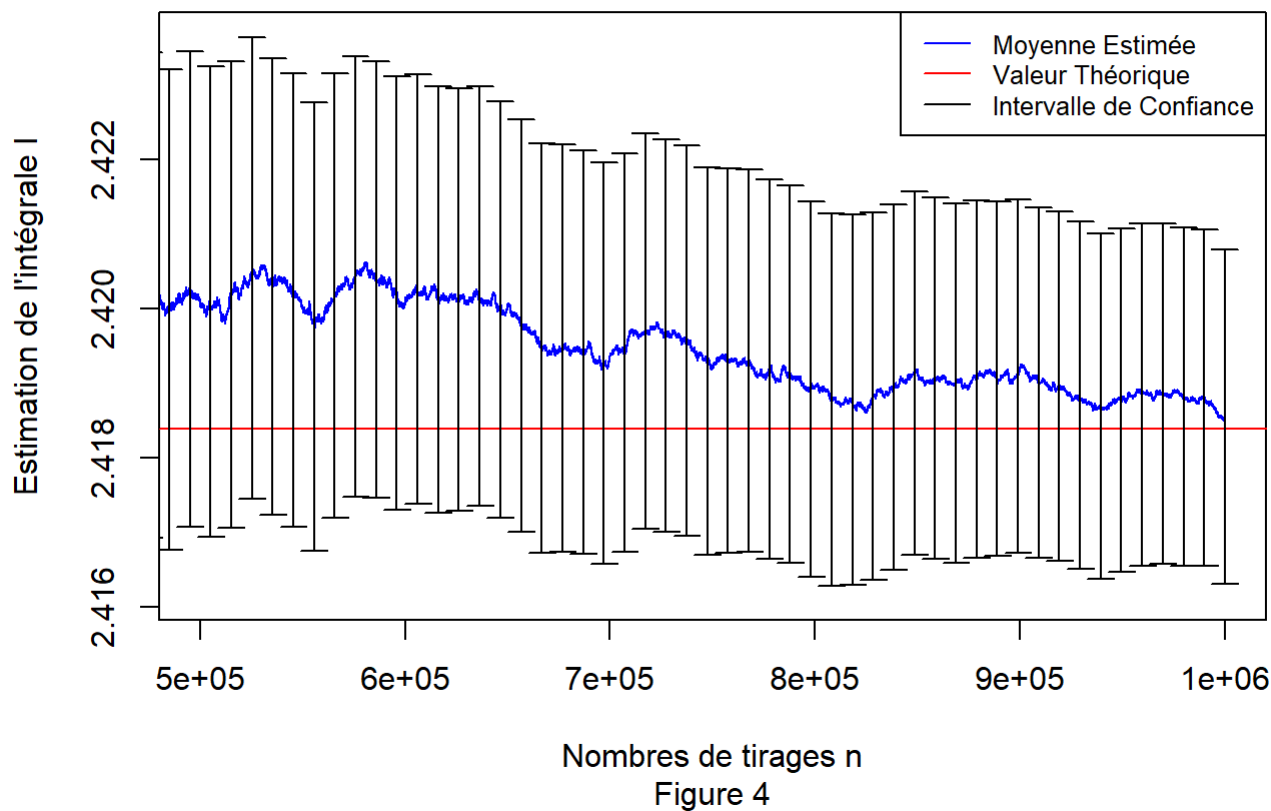
```
#Tracé Convergence de la Moyenne Estimée
plot(indices, Moyenne, type='l', col="blue",
      xlab = "Nombres de tirages n", ylab="Estimation de l'intégrale I",
      ylim=c(max(c(min((Moyenne[(0.5*n):n]-Error[(n*0.5):n])),2.2)),
             min(c(max(Moyenne[(n*0.5):n]+Error[(n*0.5):n]),2.8))),
      xlim=c(n*0.5,n),
      main="Convergence de I pour la loi de Cauchy",
      sub = "Figure 4")

#Tracé de la valeur exacte
abline (h=val_exacte$value, col="red")

#Tracé des Intervalles de Confiance (IC)
plotCI(indices_CI, Résultat_IC$Moy[indices_CI],
       uiw=Résultat_IC$Err[indices_CI],liw=Résultat_IC$Err[indices_CI],
       add=TRUE,lwd=1,pch=NA)

#Legende
legend("topright", legend = c("Moyenne Estimée", "Valeur Théorique", "Intervalle de Confiance"),
      col = c("blue", "red", "black"),
      lty = c(1, 1, 1),lwd = c(1, 1, 1), cex = 0.8)
```

Convergence de I pour la loi de Cauchy



Sur cette Figure 4, on observe que la valeur approché devient proche de la valeur exacte, qui est d'ailleurs contenue dans les intervalles de confiance. La méthode donne une précision relativement convenable, puisque pour $n=10^6$, l'erreur commise reste tolérable, pour un temps de calcul tout à fait raisonnable.

Dans les deux codes précédents, nous implémentons d'abord les fonctions nécessaires à la méthode de Monte Carlo par échantillonnage préférentiel. La fonction $g(x)$ définit l'intégrande, $g(x) = \frac{1}{1+|x|^3}$, tandis que $h(x)$ définit la fonction transformée, $h(x) = g(x)/f(x)$, où $f(x)$ est la densité de la Loi de Cauchy ($dcauchy(x)$), pour reformuler l'intégrale I en une espérance $\mathbb{E}_f[h(X)]$.

Puis, nous implémentons la fonction $IC(n)$, qui réalise le cœur du processus de simulation. Pour un n donné, cette fonction génère un n -échantillon X selon la loi de Cauchy ($rcauchy(n)$), et renvoie la suite des estimations cumulées de la moyenne $\overline{h(X)}_k$ et les marges d'erreur correspondantes pour les Intervalles de Confiance (IC) à 95%, calculées à partir de la variance empirique.

Pour la deuxième partie du code, nous cherchons à visualiser et valider cette convergence en utilisant un grand nombre de tirages ($n = 10^4$). La convergence de l'estimation de I (la courbe bleue) vers la valeur exacte (la ligne rouge, $I \approx 2.418$) est tracée. La convergence s'avère convaincante, puisque la courbe de la moyenne empirique se rapproche fortement à la valeur théorique lorsque n grandit.

De plus, les Intervalles de Confiance (IC) tracés se comportent comme attendu : ils se rétrécissent (Figure 3), et la valeur exacte I est constamment contenue dans l'intervalle lorsque n est suffisamment grand comme on peut le constater sur la Figure 4.

Ce comportement valide l'application du Théorème Central Limite (TCL) dans ce contexte d'intégration par Monte Carlo avec la densité de Cauchy. Le succès de cette simulation est en parfait accord avec l'analyse théorique initiale qui a confirmé que la condition de variance finie $\mathbb{E}_f[h(X)^2] < \infty$ était bien satisfaite.

Ensuite nous étudions la loi de Laplace.

Les mêmes vérifications s'imposent. La première, concernant l'implication $f(x) = 0 \Rightarrow g(x) = 0$ est satisfaite par le même argument. Nous devons alors vérifier que $\mathbb{E}_f[h(X)^2] < \infty$.

$$\mathbb{E}_f[h(X)^2] = \int_{\mathbb{R}} \frac{\left(\frac{1}{1+|x|^3}\right)^2}{\frac{1}{2}e^{-|x|}} dx = \int_{\mathbb{R}} \frac{2e^{|x|}}{(1+|x|^3)^2} dx$$

Pour étudier la convergence, nous examinons le comportement de l'intégrande à l'infini :

$$\frac{2e^{|x|}}{(1+|x|^3)^2} \underset{|x| \rightarrow \infty}{\sim} +\infty$$

Il n'est donc pas possible de construire un intervalle de confiance pour une densité de type Laplace et les intervalles de confiance construits formellement seraient invalides ou trompeurs.

Nous allons donc confirmer cette affirmation par le biais visuel utilisant notre programme précédent mais cette fois ci avec une densité de type laplace.

```
#code R7

g=function(x){
  return (1/(1+abs(x)^3))#fonction à intégrer
}

h=function(x){
  return(2*g(x)/exp(-abs(x))) #avec densité de la loi de Laplace pour a=1
}

#On utilise la méthode de L'inverse pour générer un n-échantillon de loi de Laplace
rlaplace=function(n){
  if(floor(n)!=n){
    stop("n doit etre un entier")
  }
  u=runif(n)
  return((u<(1/2))*log(2*u)-(u>=(1/2))*log(2*(1-u)))
}

IC=function(n){
  X=rlaplace(n)# Génération des échantillons (X) selon la Loi Laplace
  I=cumsum(h(X))/(1:n) #estimation de L'espérance E(h(X))
  Var=pmax(0,(cumsum(h(X)^2)/(1:n) -I^2)) #estimation de La variance Var(h(X))
  Err=1.96*sqrt(Var/(1:n)) # Calcul de L'Erreur
  return(list(Moy=I, Err=Err))
}
n=1e4
indices=1:n
indices_CI=seq(1, n, length.out = 100)# Définition des indices pour le tracé des IC

Résultat_IC=IC(n)

Moyenne=Résultat_IC$Moy[indices]
Error=Résultat_IC$Err[indices]

val_exacte = integrate(f = g, lower = -Inf, upper = Inf);# Calcul de la valeur exacte

cat("Valeur approchée = ")
```

```
## Valeur approchée =
```

```
print(Moyenne[length(Moyenne)]) # approximation finale
```

```
## [1] 2.413651
```

```
cat("\nValeur exacte = ")
```

```
##
## Valeur exacte =
```

```
print(val_exacte) #valeur exacte
```

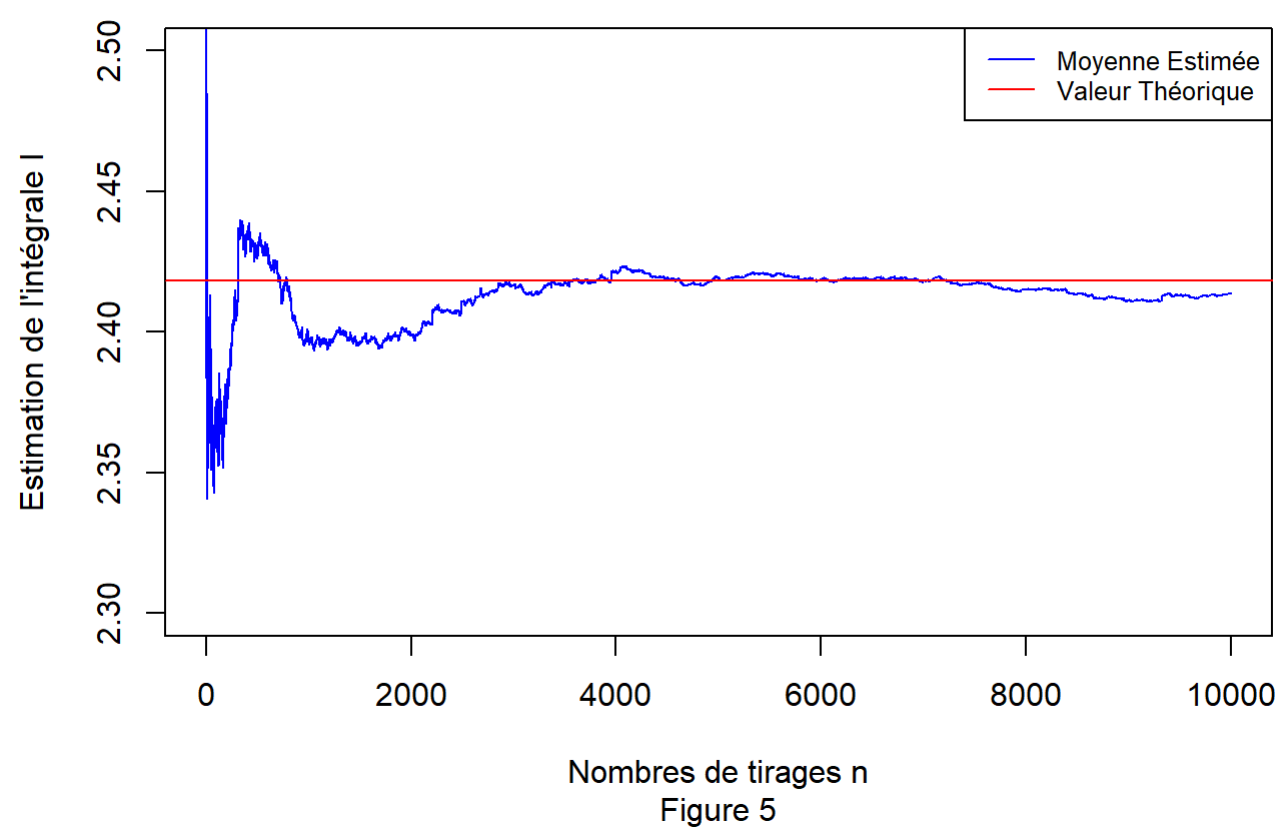
```
## 2.418399 with absolute error < 9.7e-07
```

```
#Tracé Convergence de la Moyenne Estimée
plot(indices, Moyenne, type='l', col="blue",
      xlab = "Nombres de tirages n", ylab="Estimation de l'intégrale I",
      ylim = c(2.3,2.5),
      main="Convergence de I pour la loi de Laplace",
      sub = "Figure 5")

#Tracé de la valeur exacte
abline (h=val_exacte$value, col="red")

#Legende
legend("topright", legend = c("Moyenne Estimée", "Valeur Théorique"),
      col = c("blue", "red"),
      lty = c(1, 1, 1),lwd = c(1, 1, 1), cex = 0.8)
```

Convergence de I pour la loi de Laplace



Malgré l’échec théorique du TCL, la courbe bleue de la Moyenne (Figure 5) tend toujours vers la valeur exacte $I \approx 2.418$ lorsque n est grand. Cette convergence est garantie par la Loi Forte des Grands Nombres (LFGN), qui ne requiert que $\mathbb{E}[|h(X)|] < +\infty$ (ce qui est vrai ici).

```
#code R7
g=function(x){
  return (1/(1+abs(x)^3))#fonction à intégrer
}

h=function(x){
  return(2*g(x)/exp(-abs(x))) #avec densité de la Loi de Laplace pour a=1
}

#On utilise la méthode de l'inverse pour générer un n-échantillon de loi de Laplace
rlaplace=function(n){
  if(floor(n)!=n){
    stop("n doit etre un entier")
  }
  u=runif(n)
  return((u<(1/2))*log(2*u)-(u>=(1/2))*log(2*(1-u)))
}

IC=function(n){
  X=rlaplace(n)# Génération des échantillons (X) selon la Loi q(x) = Cauchy
  I=cumsum(h(X))/(1:n) #estimation de l'espérance E(h(X))
  Var=pmax(0,(cumsum(h(X)^2)/(1:n) -I^2)) #estimation de la variance Var(h(X))
  Err=1.96*sqrt(Var/(1:n)) # Calcul de L'Erreur
  return(list(Moy=I, Err=Err))
}

n=1e6
indices=1:n
indices_CI=seq(1, n, length.out = 100)# Définition des indices pour le tracé des IC

Résultat_IC=IC(n)

Moyenne=Résultat_IC$Moy[indices]
Error=Résultat_IC$Err[indices]

val_exacte = integrate(f = g, lower = -Inf, upper = Inf);# Calcul de la valeur exacte

cat("Valeur approchée = ")

## Valeur approchée =

print(Moyenne[length(Moyenne)]) # approximation finale

## [1] 2.413021

cat("\nValeur exacte = ")
```

```
##
## Valeur exacte =
```

```
print(val_exacte) #valeur exacte
```

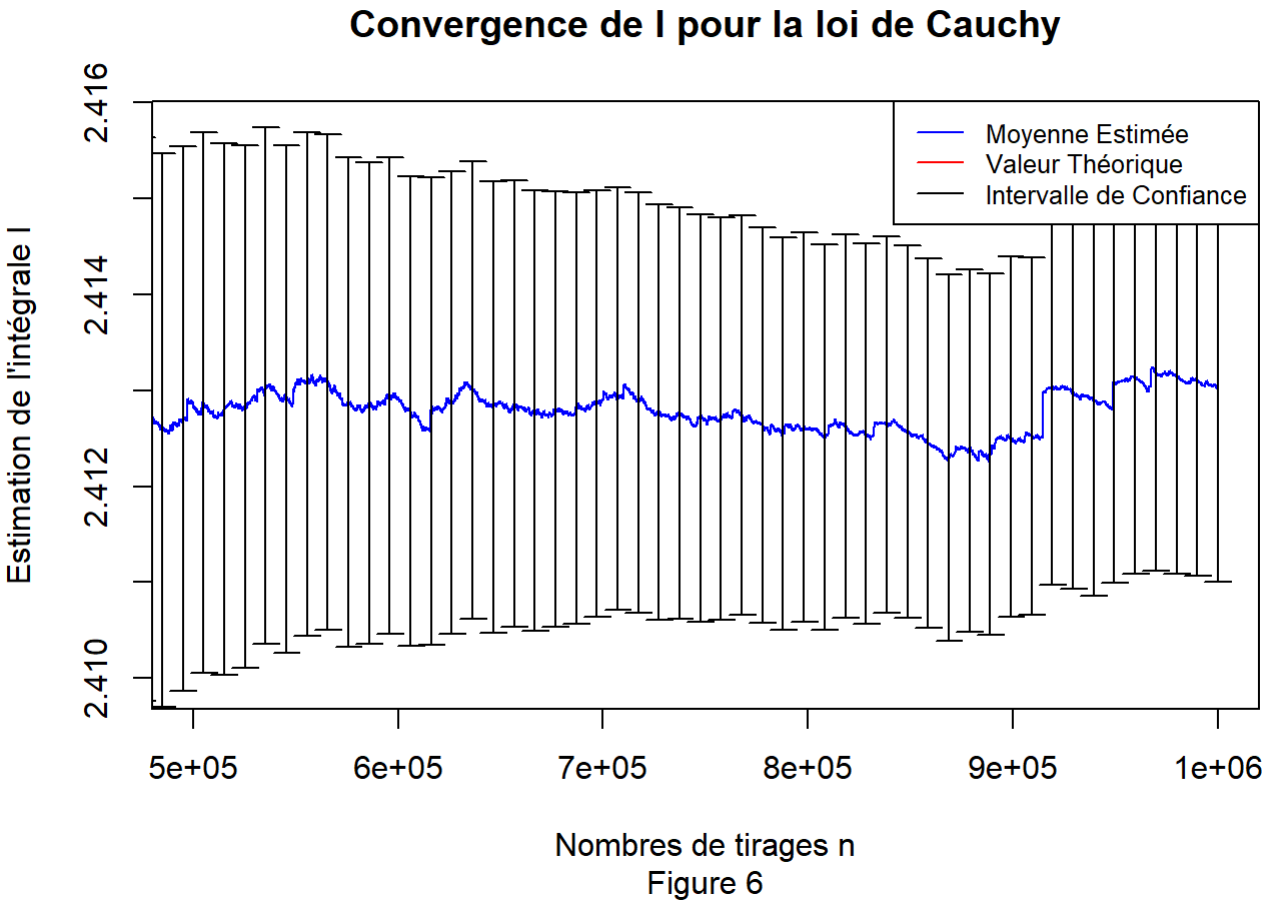
```
## 2.418399 with absolute error < 9.7e-07
```

```
#Tracé Convergence de La Moyenne Estimée
plot(indices, Moyenne, type='l', col="blue",
      xlab = "Nombres de tirages n", ylab="Estimation de l'intégrale I",
      ylim=c(max(c(min((Moyenne[(0.5*n):n]-Error[(n*0.5):n])),2.2)),
              min(c(max(Moyenne[(n*0.5):n]+Error[(n*0.5):n]),2.8))),
      xlim=c(n*0.5,n),
      main="Convergence de I pour la loi de Cauchy",
      sub = "Figure 6")

#Tracé de la valeur exacte
abline (h=val_exacte$value, col="red")

#Tracé des Intervalles de Confiance (IC)
plotCI(indices_CI, Résultat_IC$Moy[indices_CI],
        uiw=Résultat_IC$Err[indices_CI],liw=Résultat_IC$Err[indices_CI],
        add=TRUE,lwd=1,pch=NA)

#Legende
legend("topright", legend = c("Moyenne Estimée", "Valeur Théorique", "Intervalle de Confiance"),
      col = c("blue", "red", "black"),
      lty = c(1, 1, 1),lwd = c(1, 1, 1), cex = 0.8)
```



Grâce à la figure 6 on constate que la valeur exacte n’est pas toujours contenue dans l’intervalle de confiance calculé. Ce comportement irrégulier et la non-contenance de la valeur vraie, observés avec un n très grand, confirment que la méthode des IC est erronée dans ce cas, car la condition $\mathbb{E}_f[h(X)^2] < +\infty$ n’est pas respectée.

En résumé, la LFGN garantit que $\overline{h(X)}_n \xrightarrow{p.s.} I$, mais l’infinité de la variance empêche l’application du TCL, rendant la construction des intervalles de confiance invalide.

Dans cette ultime partie, nous cherchons à estimer $p = \mathbb{P}(X > 4)$ avec $X \sim \mathcal{N}(0,1)$.

Le calcul de p n’est en réalité qu’un calcul d’intégrale puisque X admet une densité :

$$p = \int_4^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = \int_{\mathbb{R}} 1_{\{x>4\}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

On pose conformément à l’équation (7a) $g(x) := 1_{]4,+\infty[}(x) \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$.

g est bien intégrable sur \mathbb{R} car $\int_{\mathbb{R}} g(x) dx = p \leq 1$ (p est une probabilité).

Pour la suite, on choisit naturellement $f(x)$ égale à la densité de la loi $N(0, 1)$.

Notons que ce choix satisfait bien l’implication $f(x) = 0 \Rightarrow g(x) = 0$. Ce qui nous permet de poser $h(x) := \frac{g(x)}{f(x)} = 1_{]4,+\infty[}(x)$.

Ainsi le calcul de p revient au calcul de l’espérance $\mathbb{E}[h(X)]$, où $X \sim N(0, 1)$.

Intéressons nous à la construction d'un interval de confiance qui nous permettrait de mesurer la qualité de notre estimation. Pour pouvoir construire un tel objet, il est nécessaire que $E[h(X)^2] < +\infty$. Cette condition est immédiatement satisfaite puisque $h(x)^2 = h(x)$.

Il nous est à présent possible d'implémenter une première estimation de p par la méthode de Monte Carlo.

```
#code R8
h=function(x){
  return(as.numeric(x>4))
}

#Fonction qui renvoie l'estimation de p et les intervalles de confiances 0.95
IC=function(n){
  X=rnorm(n) # Génération de n tirages i.i.d. N(0,1)
  I=cumsum(h(X))/(1:n) # Moyenne empirique E(h(X))
  Var=(cumsum(h(X)^2)/(1:n) -I^2) # Variance empirique Var(h(X))
  Err=1.96*sqrt(Var/(1:n)) #Calcul de L'Erreur
  return(list(Moy=I, Err=Err))
}

n=1e6
indices=1:n
indices_CI=seq(1, n, length.out = 100)

Résultat_IC=IC(n)

Moyenne=Résultat_IC$Moy
Error=Résultat_IC$Err

val_exacte = pnorm(q=4,lower.tail = F)# Calcul de la valeur exacte

cat("Valeur approchée = ")
```

```
## Valeur approchée =
```

```
print(Moyenne[length(Moyenne)]) # approximation finale
```

```
## [1] 2.8e-05
```

```
cat("\nValeur exacte = ")
```

```
##
## Valeur exacte =
```

```
print(val_exacte) #valeur exacte
```

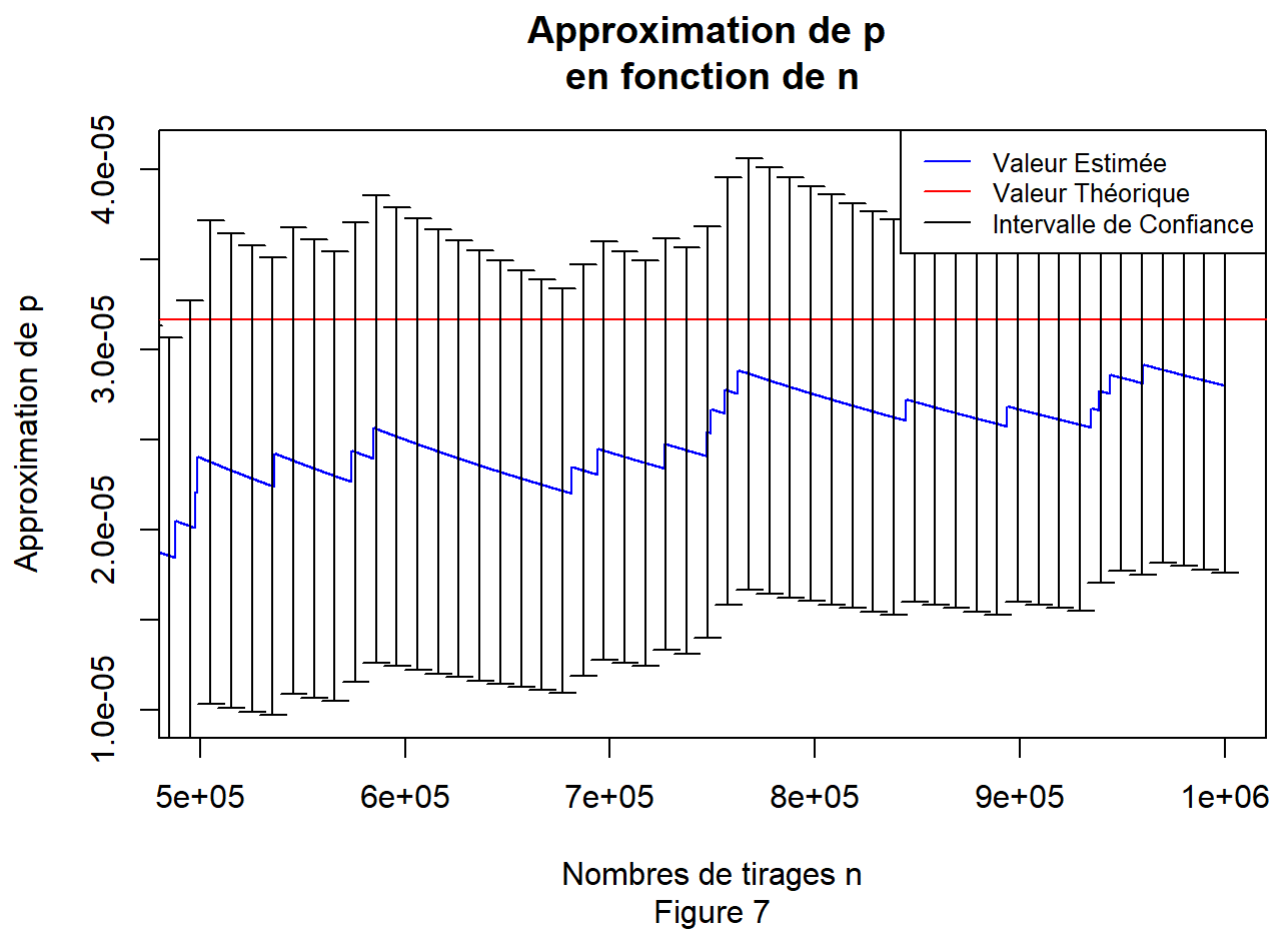
```
## [1] 3.167124e-05
```

```
plot(indices, Moyenne, type='l', col="blue",
      ylim=c(max(c(min((Moyenne[(0.5*n):n]-Error[(n*0.5):n])), -0.5)), min(c(max(Moyenne[(n*0.5):n]+Error[(n*0.5):n]), 0.5))),
      xlim=c(n*0.5,n),
      xlab = "Nombres de tirages n", ylab="Approximation de p",
      main="Approximation de p \nen fonction de n",
      sub = "Figure 7")

#Tracé de la valeur exacte
abline (h=val_exacte, col="red")

#Tracé des Intervalles de Confiance 0.95
plotCI(indices_CI, Moyenne[indices_CI],
        uiw=Error[indices_CI],liw=Error[indices_CI],
        add=TRUE,lwd=1,pch=NA)

#Legende
legend("topright", legend = c("Valeur Estimée", "Valeur Théorique", "Intervalle de Confiance"),
      col = c("blue", "red", "black"),
      lty = c(1, 1, 1),lwd = c(1, 1, 1), cex = 0.8)
```



Ce code commence par impémer la fonction $h(x)$ utilisé pour cette méthode, puis par la fonction $IC(n)$ permettant d'estimer p et de calculer les intervalles de confiance associés à ces approximations. Le code affiche ensuite la valeur approchée puis la valeur exacte, ainsi que le graphique Figure 7. Ce graphique contient l'estimation, la valeur exacte, et les intervalles de confiance.

Le constat que l'on peut faire de la Figure 7 est clair, la convergence de cette approximation est lente. On se rend compte que même pour un n très grand, il n'est pas possible d'obtenir une très bonne approximation de p en un temps raisonnable. Il est donc nécessaire de changer de méthode. Il nous ai proposé d'échantillonner suivant la loi d'une v.a. $Y := E + 4$ avec $E \sim Exp(1)$.

L'idée va donc être de prendre la fonction densité de Y comme fonction f dans (7b), on pourra alors exprimer $h(x)$ puisque $g(x) = 1_{]4,+\infty[}(x) \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ (voir début partie codeR8).

Soit $t \in \mathbb{R}$,

$$\begin{aligned} \mathbb{P}(Y \leq t) &= \mathbb{P}(E \leq t - 4) = \int_{-\infty}^{t-4} e^{-x} 1_{[0,+\infty[}(x) dx \\ &\stackrel{x=y-4}{=} \int_{-\infty}^t e^{-y+4} 1_{[4,+\infty[}(y) dy \end{aligned}$$

D'où le fait que l'on prenne $f(x) = e^{-x+4} 1_{[4,+\infty[}(x)$. Cette fonction f vérifiant bien l'implication : $f(x) = 0 \Rightarrow g(x) = 0$, on peut écrire :

$$h(x) = \frac{g(x)}{f(x)} = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2} + x - 4}$$

On va donc appliquer la méthode de Monte Carlo de manière analogue à précédemment, cependant vérifions en amont que $\mathbb{E}[h(Y)^2] < +\infty$:

$$\begin{aligned} h(y)^2 &= \frac{1}{2\pi} (e^{-\frac{y^2}{2} + y - 4})^2 = \frac{1}{2\pi} (e^{-\frac{y^2}{2} + y - 4})^2 = \frac{1}{2\pi} e^{-y^2 + 2y - 8} \\ \mathbb{E}[h(y)^2] &= \int_{\mathbb{R}} h(y)^2 f(y) dy \\ &= \int_4^{+\infty} \frac{1}{2\pi} e^{-y^2 + 2y - 8} e^{-y+4} dy \\ &= \int_4^{+\infty} \frac{1}{2\pi} e^{-y^2 + y - 4} dy < +\infty \end{aligned}$$

On peut ainsi construire des intervalles de confiance.

```
#code R8

h=function(x){
  y=-(x^2)/2 +x -4
  return(1/sqrt(2*pi)*exp(y))
}

#Fonction qui renvoie l'estimation de p et Les intervalles de confiances 0.95
IC=function(n){
  X=4+rexp(n) # Génération de n tirages i.i.d. E(1)+4
  I=cumsum(h(X))/(1:n) # Moyenne empirique E(h(X))
  Var=(cumsum(h(X)^2)/(1:n) -I^2) # Variance empirique Var(h(X))
  Err=1.96*sqrt(Var/(1:n)) #Calcul de L'Erreur
  return(list(Moy=I, Err=Err))
}

n=1e5
indices=1:n
indices_CI=seq(1,n, length.out = 100) # indices des intervalles pour Le graphique

Résultat_IC=IC(n)

Moyenne=Résultat_IC$Moy
Error=Résultat_IC$Err

val_exacte = pnorm(q=4,lower.tail = F)# Calcul de La valeur exacte

cat("Valeur approchée = ")
```

```
## Valeur approchée =
```

```
print(Moyenne[length(Moyenne)]) # approximation finale
```

```
## [1] 3.182222e-05
```

```
cat("\nValeur exacte = ")
```

```
##
## Valeur exacte =
```

```
print(val_exacte) #valeur exacte
```

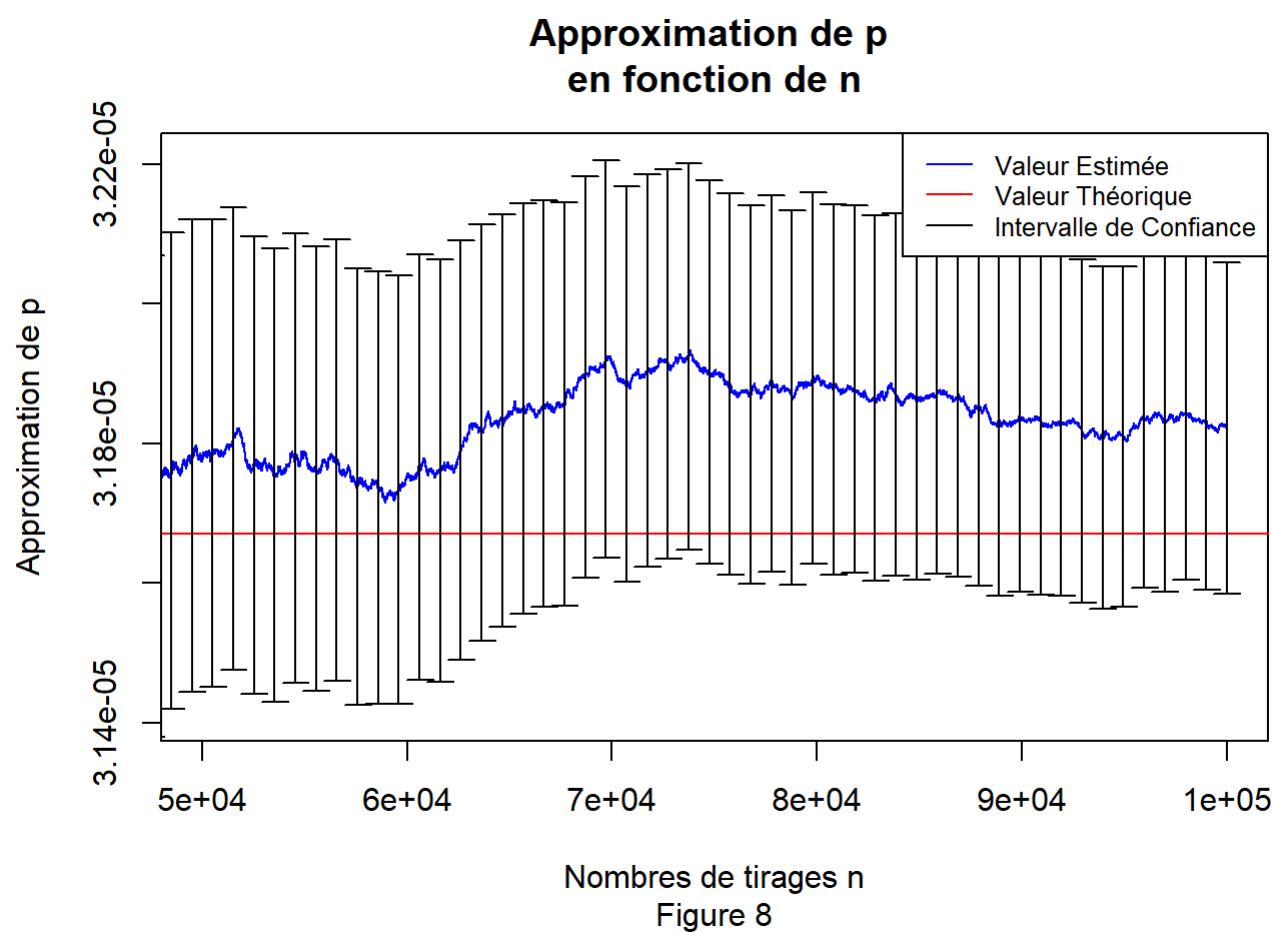
```
## [1] 3.167124e-05
```

```
plot(indices, Moyenne, type='l', col="blue",
      ylim=c(max(c(min((Moyenne[(0.5*n):n]-Error[(n*0.5):n])), -0.5)), min(c(max(Moyenne[(n*0.5):n]+Error[(n*0.5):n]), 0.5))),
      xlim=c(n*0.5,n),
      xlab = "Nombres de tirages n", ylab="Approximation de p",
      main="Approximation de p \nen fonction de n",
      sub = "Figure 8")

#Tracé de La valeur exacte
abline (h=val_exacte, col="red")

#Tracé des Intervalles de Confiance 0.95
plotCI(indices_CI, Moyenne[indices_CI],
        uiw=Error[indices_CI],liw=Error[indices_CI],
        add=TRUE,lwd=1,pch=NA)

#Legende
legend("topright", legend = c("Valeur Estimée", "Valeur Théorique", "Intervalle de Confiance"),
      col = c("blue", "red", "black"),
      lty = c(1, 1, 1),lwd = c(1, 1, 1), cex = 0.8)
```



Ce code est implémenté de façon tout à fait analogue au précédent, intégrant les changements dû à la différence d'approximation. Ainsi la fonction $h(x)$ diffère et la fonction $IC(n)$ simule selon une loi $Exp(1) + 4$.

Sur la Figure 8, cette seconde méthode de Monte Carlo semble converger plus vite vers la valeur exacte, néanmoins, les intervalles de confiance ne contiennent que rarement la valeur exacte, ce qui n'est pas le cas dans la première méthode. On peut dire par ce sens que la seconde méthode est moins "fiable" que la première. On va à présent chercher à comparer ces 2 approximations sur la taille de l'échantillon nécessaire pour obtenir une même précision, et le temps de calcul.

```
#code R8
#Fonction qui utilise la méthode utilisant une loi exponentielle
MC_exp = function(n){
  tstart = Sys.time() #debut du chronométrage de la fonction
  #fonction h utilisé dans la méthode utilisant une loi exponentielle
  h=function(x){
    y=-(x^2)/2 +x -4
    return(1/sqrt(2*pi)*exp(y))
  }

  X=4+rexp(n) # Génération de n tirages i.i.d. E(1)+4

  tend = Sys.time() #fin du chronométrage de la fonction
  return(c(mean(h(X)),tend-tstart)) # Retourne la moyenne empirique de niveau n et Le temps
}

#Fonction qui utilise la méthode utilisant une loi normale
MC_norm = function(n){
  tstart = Sys.time() #debut du chronométrage de la fonction
  #fonction h utilisé dans la méthode utilisant une loi exponentielle
  h=function(x){
    return(as.numeric(x>4))
  }

  X=rnorm(n) # Génération de n tirages i.i.d. N(0,1)

  tend = Sys.time() #fin du chronométrage de la fonction
  return(c(mean(h(X)),tend-tstart)) # Retourne la moyenne empirique de niveau n et Le temps
}

N=10^(4:8)
val_exacte = pnorm(q=4,lower.tail = F)# Calcul de la valeur exacte

#resultat pour la première méthode
resultat_norm = sapply(N,MC_norm) #approximation et temps pour différents n
resultat_norm = rbind(abs(resultat_norm[1,]-val_exacte),resultat_norm)

rownames(resultat_norm)=c("Erreur d'approximation",
                          "Approximation de p",
                          "Temps de calcul (s)")
colnames(resultat_norm)=paste0("n=", N)

cat("\nRésultat pour l'approximation utilisant une loi normale\n")
```

```
##
## Résultat pour l'approximation utilisant une loi normale
```

```
print(resultat_norm)
```

```
##              n=10000      n=1e+05      n=1e+06      n=1e+07
## Erreur d'approximation 3.167124e-05 1.671242e-06 1.671242e-06 1.728758e-06
## Approximation de p    0.000000e+00 3.000000e-05 3.000000e-05 3.340000e-05
## Temps de calcul (s)   1.532078e-03 1.184297e-02 1.249170e-01 1.594595e+00
##              n=1e+08
## Erreur d'approximation 2.512418e-07
## Approximation de p    3.142000e-05
## Temps de calcul (s)   1.583128e+01
```

```
#resultat pour la deuxième méthode
resultat_exp = sapply(N,MC_exp) #approximation et temps pour différents n
resultat_exp = rbind(abs(resultat_exp[1,]-val_exacte),resultat_exp)

rownames(resultat_exp)=c("Erreur d'approximation",
                        "Approximation de p",
                        "Temps de calcul (s)")
colnames(resultat_exp)=paste0("n=", N)

cat("\nRésultat pour l'approximation utilisant une loi exponentielle\n")
```

```
##
## Résultat pour l'approximation utilisant une loi exponentielle
```

```
print(resultat_exp)
```

```
##              n=10000      n=1e+05      n=1e+06      n=1e+07
## Erreur d'approximation 5.231829e-08 3.554389e-09 2.894522e-08 8.275977e-09
## Approximation de p    3.161892e-05 3.166769e-05 3.170019e-05 3.166297e-05
## Temps de calcul (s)   9.839535e-04 8.784056e-03 9.194493e-02 9.338751e-01
##              n=1e+08
## Erreur d'approximation 6.300886e-09
## Approximation de p    3.166494e-05
## Temps de calcul (s)   1.011136e+01
```

Ce code contient 2 fonctions analogue : **MC_exp(n)** et **MC_norm(n)** qui execute les méthodes de Monte Carlo utilisant respectivement la loi exponentielle et la loi normale. Ces fonctions renvoient l’approximation de niveau n est le temps de calcul que ctte dernière a coutée. Nous testons alors les 2 méthodes pour les puissances de 10 allant de 10^4 à 10^8 . Cela nous permet de construire un tableau contenant sur les colonnes les valeurs de n, et sur les lignes les erreur d’approximation, les valeurs d’approximations, et le temps de calcul.

Passons à l'analyse des résultats. En ce qui concerne les temps de calculs, ils sont du même ordre de grandeur qu’importe la taille de n. Cependant la précision est bien plus importante pour la seconde méthode. Tandis que la méthode utilisant une simulation de loi normale peine à éstimer p à 10^{-7} prêt, même lorsque n vaut 10^8 ; la méthode utilisant une simulation de loi exponentielle n’éprouve même pas le besoin d’un n= 10^4 pour atteindre une telle précision. Et dans le même temps que la première méthode, elle permet une précision de l’odre du milliardième !

Dans la première partie, nous avons illustré les différents modes de convergence des variables aléatoires. L’étude de \hat{b}_n a confirmé les Lois Faible et Forte des Grands Nombres (LFGN et LGFN), montrant que la moyenne empirique converge de manière fiable vers l’espérance théorique.

L’analyse du maximum cumulé M_n (Code R2) a validé la convergence presque sûre. Le contre-exemple de la suite de Bernoulli $X_n \sim \text{Ber}(1/\sqrt{n})$ (Code R3) a servi à émontrer que la convergence en probabilité est une condition plus faible qui n’implique pas nécessairement la convergence presque sûre.

Finalement, l’illustration du Théorème Central Limite (TCL) sur la loi du χ^2 (Code R5) a justifié l’usage de la loi normale pour la construction des intervalles de confiance.

La seconde partie, dédiée à l’intégration par les méthodes de Monte Carlo (MC), a établi une règle essentielle : l’efficacité et la validité des estimations dépendent de la finitude de la variance $\mathbb{E}_f[h(X)^2]$.

L’étude avec la densité de Cauchy (Code R7) a montré que lorsque la variance est finie, les intervalles de confiance (IC) fonctionnent parfaitement et se rétrécissent comme l’exige le TCL.

Inversement, l’analyse avec la densité de Laplace (Code R7) a démontré que, même si la LFGN garantie la convergence de l’estimation vers la valeur exacte, l’infinité de la variance empêche l’application du TCL, rendant les IC invalides ou trompeurs.

Enfin, pour estimer la probabilité $\mathbb{P}(X > 4)$, nous avons prouvé que l’Échantillonnage Préférentiel (Code R8) est très avantageux. En ajustant la loi de tirage (Loi Exponentielle décalée), nous avons réduit la variance de l’estimateur, obtenant ainsi une précision bien supérieure à celle de la méthode standard.