M1 IM - Simulation stochastique (Monte-Carlo) - 2017-2018
`http://math.unice.fr/~rubentha/cours.html`

Corrigé du contrôle no 3, sujet B (durée 2h)

Documents et calculatrices interdits. La plus grande importance a été accordée lors de la correction à la justification des réponses.

Exercice 1.

(1) Soit φ une fonction de C_b^+. Nous calculons

$$\begin{aligned} \mathbb{E}(\varphi(\mathtt{t})) &= \mathbb{E}(\varphi(-\log(U)/\lambda)) \\ &= \int_{[0;1]} \varphi(-\log(u)/\lambda)du \\ (u=e^{-\lambda x}) &= \int_{-\infty}^{0} \varphi(x)(-\lambda)e^{-\lambda x}dx \\ &= \int_0^{+\infty} \varphi(x)\lambda e^{-\lambda x}dx\,. \end{aligned}$$

Donc la variable est de loi $\mathcal{E}(\lambda)$.

(2) À chaque passage dans la boucle, la probabilité de mettre `b` à 1 est $1-e^{-\lambda}$ (puisque le `t` est de la loi $\mathcal{E}(\lambda)$). Du coup, la loi de `n` (=nombre de boucles) est $\mathcal{G}(1-\lambda)$.

(3) Le nombre moyen de boucle est l'espérance de cette variable :

$$\begin{aligned} \sum_{n=1}^{+\infty} n(1-e^{-\lambda})(e^{-\lambda})^{n-1} &= (1-e^{-\lambda})\left(\frac{d}{dz}\sum_{n=0}^{+\infty} z^n\right)_{z=e^{-\lambda}} \\ &= (1-e^{-\lambda})\left(\frac{1}{(1-z)^2}\right)_{z=e^{-\lambda}} \\ &= \frac{1}{1-e^{-\lambda}}\,. \end{aligned}$$

Exercice 2.

(1) Soit φ dans C_b^+. Nous calculons :

$$\begin{aligned} \mathbb{E}(\varphi(2\tan(\pi U^2/2))) &= \int_0^1 \varphi(2\tan(\pi u^2/2))\times 2udu \\ (x=2\tan(\pi u^2/2)) &= \int_0^{+\infty} \varphi(x)2\sqrt{\frac{2}{\pi}\arctan\left(\frac{x}{2}\right)}\frac{1}{2\sqrt{\frac{2}{\pi}\arctan\left(\frac{x}{2}\right)}}\frac{2}{\pi}\frac{1}{2}\frac{1}{\left(1+\left(\frac{x}{2}\right)^2\right)}dx \\ &= \int_0^{+\infty} \varphi(x)\frac{1}{\pi}\frac{1}{1+\frac{x^2}{4}}dx\,. \end{aligned}$$

D'où le résultat.

(2) Pour $x \geq 0$, nous avons

$$\begin{aligned} f(x) &\leq \frac{1}{Z\left(1+\frac{x^2}{4}\right)} \\ &= \frac{\pi}{Z}\times g(x)\,. \end{aligned}$$

Nous prenons donc $C=\pi/Z$.

(3) Nous utilisons la méthode d'acceptation-rejet basée sur l'inégalité ci-dessus. Nous remarquons que, pour une variable U de loi $\mathcal{U}([0;1])$ et X de loi de densité g, nous devons tester

$$U \leq \frac{f(X)}{Cg(X)} = \frac{(\sin x)^2}{(1+x^2)}\times\pi\left(1+\frac{x^2}{4}\right)\,.$$

Donc nous n'avons pas besoin de la constante Z.
Pour simuler une variable de densité g, nous commençons par calculer sa fonction de répartition (pour $x \geq 0$)

$$\begin{aligned} G(x) &= \int_0^x \frac{1}{\pi\left(1+\frac{t^2}{4}\right)} dt \\ &= \left[\frac{2}{\pi}\arctan\left(\frac{t}{2}\right)\right]_0^x \\ &= \frac{2}{\pi}\arctan\left(\frac{x}{2}\right) . \end{aligned}$$

Nous calculons ensuite le pseudo-inverse de G. Pour $u \in [0;1]$,

$$G^{-1}(u) = 2\tan\left(\frac{\pi u}{2}\right) .$$

Voir le programme dans le cadre algorithme 1.

Algorithme 1 Acceptation/rejet

```
# simulation de variable de densité g
simu<-function(t)
{
  u=runif(1,0,1)
  return(2*tan(pi*u/2))
}
# acceptatio/rejet
b=0
while (b==0)
{
  x=simu(1)
  v=runif(1,0,1)
  if (v<(sin(x))^2*pi*(1+x^2/4)/(1+x^2))
  {b=1}
}
print(x)
```

Exercice 3.

(1) Les termes de la série sont positifs. Nous avons

$$Z = 1 + \sum_{x\in N^*} \frac{2}{1+\beta x^4} .$$

Comme $x^2 \times \frac{2}{1+\beta x^4} \underset{x\to+\infty}{\longrightarrow} 0$, nous avons $Z < \infty$ (comparaison avec une série de Riemann).

(2) Si x et y dans $\mathbb{Z}$ avec $x < y$, avec $x - y$ pair, nous avons

$$Q(x,y) \geq Q(x,x+2)Q(x+2,x+4)\dots Q(y-2,y) > 0 .$$

(3) Si x et y dans $\mathbb{Z}$ avec $x < y$, avec $x - y$ impair, nous avons

$$Q(x,y) \geq Q(x,x+2)Q(x+2,x+4)\dots Q(y-1,y+1)Q(y+1,y+3)Q(y+3,y) > 0 .$$

(La démonstration pour $x > y$ est très similaire.) Donc Q est irréductible.

(4) Si nous sommes en x et que nous proposons y, le ratio d'acceptation est

$$\begin{aligned}\alpha(x,y) &= \min\left(1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right) \\ &= \min\left(1, \frac{(1+\beta x^4)Q(y,x)}{(1+\beta y^4)Q(x,y)}\right) \\ &= \min\left(1, \frac{1+\beta x^4}{1+\beta y^4}\right) ,\end{aligned}$$

car Q est symétrique. Donc nous n'avons pas besoin de la constante Z. Voir l'algorithme 2 pour le programme.

Algorithme 2 Metropolis

```
# noyau de Markov
qmarkov<-function(x)
{
  u=runif(1,0,1)
  if (u<1/3)
  {z=x+2}
  else if (u<1/2)
  {z=x+3}
  else if (u<5/6)
  {z=x-2}
  else
  {z=x-3}
  return(z)
}
# Metropolis
beta=0.5
n=100
x=0
print(x)
for (i in 1:n)
{
  y=qmarkov(x)
  alpha=min(1,(1+beta*x^4)/(1+beta*y^4))
  w=runif(1,0,1)
  if (w<alpha)
  { x<-y }
  print(x)
}
```