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

Corrigé du contrôle no 3, sujet A (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) À chaque fois, `x` vaut 1 avec probabilité p $(= 0,2)$ et 0 avec probabilité $1-p$. Donc `x` est de loi $\mathcal{B}(p)$ (loi de Bernoulli).

(2) Dans la boucle, on additionne $n = 10$ variables de Bernouilli de paramètre p (indépendantes). Donc le résultat est de loi $\mathcal{B}(n,p)$ (loi binomiale).

(3) On fait toujours exactement 10 boucles.

Exercice 2.

(1) Nous remarquons que X et $-X$ ont même loi donc nous avons la convergence :

$$\frac{1}{2n}\sum_{i=1}^{n}\left[e^{1/X_i}\mathbb{1}_{|X_i|>1} + e^{-1/X_i}\mathbb{1}_{|-X_i|>1}\right] \xrightarrow[n\to+\infty]{\text{p.s.}} I\,,$$

parce que l'hypothèse L^1 est bien vérifiée :

$$\begin{aligned}\mathbb{E}(|e^{1/X}\mathbb{1}_{|X|>1}|) &= \int_{]-\infty;-1]\cup[1;+\infty[} e^{1/x}e^{-x^2/2}dx \\ &\leq \int_{]-\infty;-1]\cup[1;+\infty[} e^{-x^2/2}dx \\ &\leq \int_{\mathbb{R}} e^{-x^2/2}dx \\ &= 1\,.\end{aligned}$$

(2) Voir Algorithme 1

Algorithme 1 Variables antithétiques

```
n=1000
s=0
for (i in 1:n)
{
  x=rnorm(1,0,1)
  if (abs(x)>1)
  { s=s+(exp(1/x)+exp(-1/x))/2 }
}
print(s/n)
```

Exercice 3.

(1) Voir Algorithme 2.

(2) Nous remarquons que le taux d'acceptation, si on est en u et que l'on propose v, est :

$$\begin{aligned}\alpha(u,v) &= \min\left(1, \frac{\pi(v)Q(v,u)}{\pi(u)Q(u,v)}\right) \\ &= \min\left(1, \frac{c(v)}{c(u)}\right)\,,\end{aligned}$$

car Q est symétrique. Nous obtenons donc l'algorithme 3.

Algorithme 2 Calcul de la fonction c.

```
orbite<-function(u,k)
{
  s=0
  b=0
  x=k
  while (b==0)
  {
    x=u[x]
    s=s+1
    if (x==k)
    {b<-1}
  }
  return(s)
}
n=10
fc<-function(u)
{
  s=0
  for (i in 1:n)
  {
    if orbite(u,i)>s
    {s=orbite(u,i)}
  }
  return(s)
}
```

Algorithme 3 Metropolis

```
n=10
N=100
u=c()
# on commence par remplir u
for (i in 1:n)
{
  u=c(u,i)
}
# on simule ensuite la chaîne sur N pas
print(u)
for (i in 1:N)
{
  #proposition
  i<-1
  j<-1
  while (i==j)
  {
    i=floor(runif(1,0,n))+1
    j=floor(runif(1,0,n))+1
  }
  v=u
  v[i]=u[j]
  v[j]=u[i]
  #calcul du ratio
  alpha=min(1,orbite(v)/orbite(u))
  # acceptation/rejet
  w=runif(1,0,1)
  if (w<alpha)
  {u=v}
  print(u)
}
```