

TP 2 - MÉTHODES DE MONTE-CARLO - CORRIGÉ SUCCINCT

Exercice 1. 1. L'intégrale I_1 est l'aire du disque unité de \mathbb{R}^2 , I_2 aussi, et I_3 le volume de la boule unité dans \mathbb{R}^3 . Les intégrales I_1 et I_2 valent donc π , et I_3 vaut $4/3\pi$ (et donc $\pi = 3I_3/4$). On peut écrire

$$I_1 = 4\mathbb{E}[\sqrt{(1-U^2)}] \text{ avec } U \sim \mathcal{U}_{[0,1]},$$

$$I_2 = 4\mathbb{E}[\mathbf{1}_{U_1^2+U_2^2 \leq 1}] \text{ avec } U_1, U_2 \sim \mathcal{U}_{[-1;1]}, U_1 \text{ indépendante de } U_2.$$

$$I_3 = 8\mathbb{E}[\mathbf{1}_{U_1^2+U_2^2+U_3^2 \leq 1}] \text{ avec } U_1, U_2, U_3 \sim \mathcal{U}_{[-1;1]}, U_1, U_2, U_3 \text{ indépendantes.}$$

2. Voici les trois fonctions demandées, enregistrées dans un fichier `TP2Ex1CalculPi.sci`. On approche π en utilisant les formules

$$I_{N,1} = \frac{4}{N} \sum_{i=1}^N \sqrt{1-U_i^2}, \quad I_{N,2} = \frac{4}{N} \sum_{i=1}^N \mathbf{1}_{U_{1,i}^2+U_{2,i}^2 \leq 1}, \quad I_{N,3} = \frac{8}{N} \sum_{i=1}^N \mathbf{1}_{U_{1,i}^2+U_{2,i}^2+U_{3,i}^2 \leq 1}.$$

```
function pi1=TP2Ex1CalculPi_MC1(n)
```

```
    U=rand(1,n);
```

```
    pi1=mean(4*sqrt(1-U.^2));
```

```
endfunction
```

```
function pi2=TP2Ex1CalculPi_MC2(n)
```

```
    U1=2*rand(1,n)-1;
```

```
    U2=2*rand(1,n)-1;
```

```
    pi2=mean(4*bool2s((U1.^2+U2.^2)<=1));
```

```
endfunction
```

```
function pi3=TP2Ex1CalculPi_MC3(n)
```

```
    U1=2*rand(1,n)-1;
```

```
    U2=2*rand(1,n)-1;
```

```
    U3=2*rand(1,n)-1;
```

```
    pi3=2*3*mean(bool2s((U1.^2+U2.^2+U3.^2)<=1));
```

```
endfunction
```

3. Soit toujours $I_{N,1} = \sum_{i=1}^N 4\sqrt{1-U_i^2}/N$, pour $\{U_1, \dots, U_n\}$ un échantillon de variables $\mathcal{U}_{[0,1]}$. On cherche N tel que

$$\mathbb{P}(|I_{N,1} - \pi| \leq 10^{-2}/2) \geq 1 - \alpha,$$

avec $\alpha = 0.05$.

Le Théorème Central limite et la méthode appliquée dans l'Exercice 5 entraînent

$$\lim_{N \rightarrow \infty} \mathbb{P} \left(|I_{N,1} - \pi| \leq q_{1-\alpha/2} \sqrt{\frac{\text{Var}(4\sqrt{1-U_1^2})}{N}} \right) \geq 1 - \alpha.$$

En minorant grossièrement π (inconnu!) par 3,

$$\text{Var} \left(4\sqrt{1-U_1^2} \right) = \frac{32}{3} - \pi^2 \leq \frac{5}{3},$$

et donc on cherche N tel que

$$2\sqrt{\frac{5/3}{N}} q_{1-\alpha/2} \leq 10^{-2}.$$

On choisit donc $N \geq 4 \times 10^4 \times q_{1-\alpha/2}^2 \times 5/3$. On peut "vérifier" dans Scilab que pour un N plus grand que ce seuil l'écart entre $I_{N,1}$ et π est bien inférieur à 10^{-2} (voir le début du script à la question suivante).

4. Voici un script illustrant la convergence des trois méthodes en représentant les approximations $I_{N,i}$ en fonction de N .

```
//Script TP2 Ex 1
clear
exec('TP2Ex1CalculPi.sci')

n=1000
pi1=TP2Ex1CalculPi_MC1(n)
pi2=TP2Ex1CalculPi_MC2(n)
pi3=TP2Ex1CalculPi_MC3(n)

N=floor(4*10^4*cdfnor('X',0,1,0.975,0.025)^2*5/3)+1; //valeur calculee Question 2.
pi1bis=TP1Ex6CalculPi_MC1(n)
abs(pi1bis-%pi)

//Illustration de la convergence des 3 methodes

vect_n=[100:500:10000];
vect_pi1=[];
vect_pi2=[];
vect_pi3=[];
for n=vect_n
    vect_pi1=[vect_pi1,TP2Ex1CalculPi_MC1(n)];
    vect_pi2=[vect_pi2,TP2Ex1CalculPi_MC2(n)];
    vect_pi3=[vect_pi3,TP2Ex1CalculPi_MC3(n)];
end
scf(1)
clf
plot2d(vect_n,[vect_pi1',vect_pi2',vect_pi3'],style=[1,2,3],...
    rect=[0,3,10000,3.3])
plot2d([min(vect_n),max(vect_n)],[%pi,%pi],style=5,rect=[0,3,10000,3.3])
legend(['MC1','MC2','MC3','vraie_valeur']);
xtitle('Approximation de '+string(%pi)+' par Monte Carlo','n','')
```

Le graphe obtenu est représenté à la Figure 1 ci-dessous. On peut aussi illustrer plus en détail la convergence en représentant $I_{N,l}$ ($l = 1, 2, 3$) en fonction de N , mais aussi les bornes de l'intervalle de confiance obtenu par Théorème Central limite pour chaque valeur de N . A titre d'exemple, pour $I_{N,1}$, on ajoute donc la fonction suivante au fichier TP2Ex1CalculPi.sci.

```
function [piInf, piMoy, piSup]=TP2Ex1CalculPi_MC1bis(n)
    //meme principe que la 1ere fonction mais retourne les approximations
    //obtenues pour toute valeur <=n, et les bornes de l'ICA associe
    U=rand(1,n);
    piMoy=cumsum(4*sqrt(1-U.^2))./[1:n];
    deviation=1.96*sqrt(5/3)./sqrt([1:n]);
    piInf=piMoy-deviation;
    piSup=piMoy+deviation;
endfunction
```

et on peut l'utiliser en complétant le script précédent par

```
//Illustration de la convergence pour la premiere methode,
//avec Bornes de l'IC correspondant (obtenu par TCL)

n=10000;
[piInf, piMoy, piSup]=CalculPi_MC1bis(n)
scf(2)
clf
```

```
plot2d([1:n]', [piInf', piMoy', piSup'], style=[1,3,1], rect=[0,3,10000,3.3])
plot2d([min(vect_n), max(vect_n)], [%pi, %pi], style=5, rect=[0,3,10000,3.3])
xlabel('Approximation de '+string(%pi)+' par Monte Carlo1, et bornes d''ICA', 'n', '')
```

ce qui permet d'obtenir la seconde partie de la Figure 1.

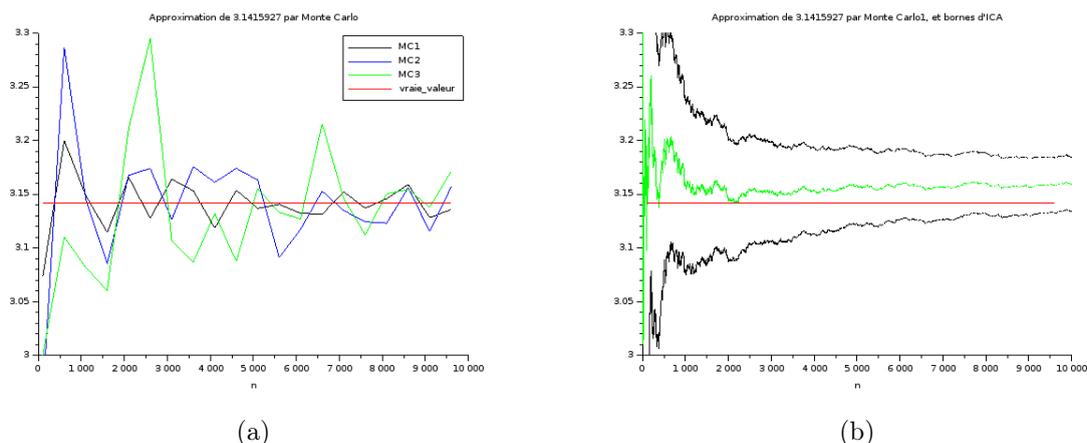


FIGURE 1 – Exercice 1. Approximation de π par Monte-Carlo. (a) Convergence des 3 approximations $I_{N,l}$, $l = 1, 2, 3$. (b) Convergence de $I_{N,1}$ et borne de l'intervalle de confiance de niveau 95%.

Exercice 2. 1. Les variables X et $1 - X$ ont même loi. Donc $I = \mathbb{E}[f(X)] = \mathbb{E}[f(1 - X)]$, puis $I = \mathbb{E}\left[\frac{1}{2}(f(X) + f(1 - X))\right]$.

2. En utilisant l'expression de la variance d'une somme, le fait que X et $1 - X$ ont même loi, et l'Inégalité de Cauchy-Schwarz,

$$\begin{aligned} \text{Var}(g(X)) &= \frac{1}{4} (\text{Var}(f(X)) + \text{Var}(f(1 - X)) + 2\text{Cov}(f(X), f(1 - X))), \\ &= \frac{1}{2} (\text{Var}(f(X)) + \text{Cov}(f(X), f(1 - X))) \leq \text{Var}(f(X)). \end{aligned}$$

3.

```
// Script TP2 Ex2
n=1000;
vraie_valeur=exp(1)-1;

u1=rand(1,n);
approx1=mean(exp(u1));
erreur1=abs(approx1-vraie_valeur)
disp('Erreur avec Monte-Carlo classique:'+string(erreur1))

u2=rand(1,n);
approx2=mean((1/2)*(exp(u2)+exp(1-u2)));
erreur2=abs(approx2-vraie_valeur)
disp('Erreur avec reduction de variance:'+string(erreur2))
```

Exercice 3. Rappel : si $X \sim \mathcal{P}(\lambda)$, $\mathbb{E}[X] = \text{Var}(X) = \lambda$.

1. Comme les variables $(X_i)_i$ sont *i.i.d.*, L^2 , d'espérance et de variance λ , le Théorème Central limite prouve que

$$\sqrt{n}(\bar{X}_n - \lambda) \xrightarrow{\mathcal{L}} Z_\lambda \sim \mathcal{N}(0, \lambda). \quad (1)$$

Par ailleurs, la Loi faible des grands nombres justifie que $(\bar{X}_n)_n$ converge en probabilité vers $\mathbb{E}[X_1] = \lambda$. Donc, comme la fonction $x \mapsto \sqrt{x}$ est continue,

$$\sqrt{\bar{X}_n} \xrightarrow{\mathbb{P}} \sqrt{\lambda}. \quad (2)$$

Donc par le Lemme de Slutsky, en tenant compte de (1) et (2),

$$\frac{\sqrt{n}(\bar{X}_n - \lambda)}{\sqrt{\bar{X}_n}} \xrightarrow{\mathcal{L}} \frac{Z_\lambda}{\sqrt{\lambda}} = N \sim \mathcal{N}(0, 1).$$

Pour bâtir $\hat{I}_{n,1}(\alpha)$, on peut donc chercher q tel que

$$\lim_{n \rightarrow \infty} \mathbb{P} \left(\sqrt{n} \left| \bar{X}_n - \frac{\lambda}{\sqrt{\bar{X}_n}} \right| \geq q \right) = \mathbb{P}(|N| \geq q) \leq \alpha.$$

Il suffit de prendre $q = q_{1-\alpha/2}$ le quantile d'ordre $1 - \alpha/2$ de la loi $\mathcal{N}(0, 1)$. Donc,

$$\hat{I}_{n,1}(\alpha) = \left[\bar{X}_n - \frac{q_{1-\alpha/2}}{\sqrt{n}} \sqrt{\bar{X}_n}; \bar{X}_n + \frac{q_{1-\alpha/2}}{\sqrt{n}} \sqrt{\bar{X}_n} \right].$$

Remarque : On définit la variance empirique par $S_n^2 = \sum_{i=1}^n (X_i - \bar{X}_n)^2 / (n-1)$. On peut alors montrer que l'on a aussi la convergence de $(S_n^2)_n$ vers $\text{Var}(X_1) = \lambda$ en probabilité. On peut alors appliquer la même méthode : en divisant (1) par $\sqrt{S_n^2}$, on obtient une nouvelle convergence en loi vers N , et on en déduit un second intervalle de confiance asymptotique :

$$\hat{I}_{n,1}^{bis}(\alpha) = \left[\bar{X}_n - \frac{q_{1-\alpha/2}}{\sqrt{n}} \sqrt{S_n^2}; \bar{X}_n + \frac{q_{1-\alpha/2}}{\sqrt{n}} \sqrt{S_n^2} \right].$$

2. Par Loi faible des grands nombres et continuité de la fonction $x \mapsto \sqrt{x} + \sqrt{\lambda}$,

$$\sqrt{\bar{X}_n} + \sqrt{\lambda} \xrightarrow{\mathbb{P}} 2\sqrt{\lambda}.$$

En utilisant ceci et (1), on peut à nouveau appliquer le Lemme de Slutsky pour obtenir

$$2\sqrt{n} \left(\sqrt{\bar{X}_n} - \sqrt{\lambda} \right) = 2\sqrt{n} \frac{(\bar{X}_n - \lambda)}{\sqrt{\bar{X}_n} + \sqrt{\lambda}} \xrightarrow{\mathcal{L}} \frac{2Z_\lambda}{2\lambda} = N \sim \mathcal{N}(0, 1).$$

Comme ci-dessus, on en déduit un intervalle de confiance asymptotique pour $\sqrt{\lambda}$, puis, pour n assez grand, un intervalle de confiance asymptotique pour λ :

$$\hat{I}_{n,2}(\alpha) = \left[\left(\sqrt{\bar{X}_n} - \frac{q_{1-\alpha/2}}{2\sqrt{n}} \right)^2; \left(\sqrt{\bar{X}_n} + \frac{q_{1-\alpha/2}}{2\sqrt{n}} \right)^2 \right].$$

3. Soit un rang n fixé. On cherche pour chaque intervalle \hat{I}_n à approcher par Monte-Carlo l'espérance de Y , avec $Y = \mathbf{1}_{\lambda \in \hat{I}_n}$. On doit donc simuler un N -échantillon (Y_1, \dots, Y_N) et en prendre la moyenne empirique. Simuler une variable Y , c'est déjà simuler un n -échantillon de variables $X_i \sim \mathcal{P}(\lambda)$. On doit donc, pour chaque valeur de n , simuler un tableau $n \times N$ de variables de loi $\mathcal{P}(\lambda)$.

```
// Script TP2 Ex 3
```

```
clear
```

```
vect_n=[10,30,50,100,500] //rangs auxquels on teste les niveaux des IC
```

```
N=1000 //nb de repetition pour le Monte-Carlo
```

```
lambda=3; //parametre de la loi de Poisson
```

```
quantile=cdfnorr('X',0,1,0.975,0.025) //quantile d'ordre 1-alpha/2 de N(0,1)
```

```
//avec alpha=0.05
```

```
//vecteurs qui vont contenir les niveaux des IC pr les valeurs de n dans vect_n
niveauIC1=[]
niveauIC2=[]
niveauIC1bis=[]

for n=vect_n
    Xech=grand(n,N,'poi',lambda)
    Xbar=mean(Xech,'r') //echantillon de taille N de X_nbar (moy empirique)
    S2=variance(Xech,'r') //echantillon de taille N de S2 (var empirique)

    //niveau de l'IC1
    Y1ech=((Xbar-quantile*sqrt(Xbar)/sqrt(n))<lambda)...
        &((Xbar+quantile*sqrt(Xbar)/sqrt(n))>lambda);
    niveauIC1=[niveauIC1,mean(1*Y1ech)]

    //niveau de l'IC1bis
    Y1bisech=((Xbar-quantile*sqrt(S2)/sqrt(n))<lambda)...
        &((Xbar+quantile*sqrt(S2)/sqrt(n))>lambda);
    niveauIC1bis=[niveauIC1bis,mean(1*Y1bisech)]

    //niveau de l'IC2
    Y2ech=((sqrt(Xbar)-quantile/(2*sqrt(n))).^2<lambda)...
        &(((sqrt(Xbar)+quantile/(2*sqrt(n))).^2)>lambda);
    niveauIC2=[niveauIC2,mean(1*Y2ech)]
end
disp(niveauIC1)
disp(niveauIC1bis)
disp(niveauIC2)
```