Université de Poitiers
Département de Mathématiques

Processus à temps discret
M1 MFA–MMAS, année 2013–2014

TRAVAUX PRATIQUES. — FEUILLE N° 1

Le propos de ces travaux pratiques est de mettre en œuvre quelques techniques d'estimation numérique reposant directement sur la loi forte des grands nombres et du théorème central limite. En termes de processus, ce sont essentiellement des suites de variables aléatoires indépendantes et identiquement distribuées que nous regarderons.

Actuellement, un grand nombre de méthodes numériques utilisant le hasard sont qualifiées de méthodes de Monte Carlo (cette terminologie date des travaux de S. Ulam et N. Metropolis), si bien qu'on s'y perd facilement. Plus loin, il s'agira de méthodes d'intégration de Monte Carlo (la plus simple et la plus commune).

1. **L'aiguille de Buffon**

En 1777, Buffon[1] publie une note au sujet du « jeu du franc carreau ». Il s'agissait d'un jeu alors très en vogue à la cour consistant à jeter à tour de rôle une pièce d'un écu ; le lancer est gagnant si la pièce ne chevauche pas le réseau formé par le carrelage. Dans cette note, Buffon étudie un problème similaire où la pièce est remplacée par une aiguille et les carreaux par des lattes de parquet.

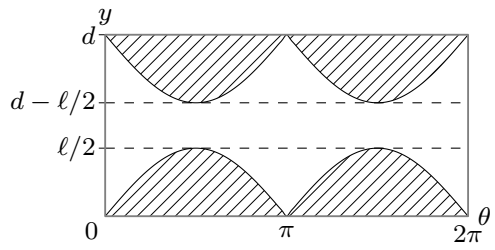
On dispose d'un aiguille, ou d'un bâton, de longueur ℓ (dont le diamètre est négligé). On la lance « au hasard » sur un parquet en bois formant un réseau régulier de droites parallèles d'égale distance d avec $\ell < d$. La question est de savoir quelle est la probabilité que l'aiguille intersecte l'une de ces droites.

La difficulté de ce problème tient à la modélisation. L'aiguille doit se répartir uniformément sur le plan — implicitement infini — de lancer. Il est donc nécessaire de faire quelques réductions pour munir le modèle (qui doit décrire comment tombe l'aiguille) d'une mesure de probabilité.

Si M désigne le centre de l'aiguille, sa coordonnée parallèle aux lattes peut être ignorée. Sa coordonnée orthogonale aux lattes est supposée de « loi » uniforme sur $\mathbb{R}$; ainsi, modulo d, cette coordonnée Y est de loi uniforme sur $[0, d]$. L'aiguille pouvant avoir un bout pointu et un autre qui ne l'est pas, désignons par Θ à valeurs dans $[0, 2\pi[$ l'angle formé par le bout pointu et une des deux directions des lattes, la variable Θ est supposée de loi uniforme.

On constate qu'il ne peut y avoir intersection que si

$$\begin{cases} Y \leqslant |\sin\Theta| \times \ell/2 \\ \text{ou} \\ Y \geqslant d - |\sin\Theta| \times \ell/2. \end{cases}$$



En représentant les coordonnées (θ, y) dans $[0, 2\pi] \times [0, d]$ muni de la mesure de probabilité uniforme (loi du couple (Θ, Y)), tout revient à calculer 2 fois l'aire normalisée sous la courbe

1. Georges Louis Leclerc, comte de Buffon (1707–1788), naturaliste français. Le problème fut posé en 1733 ([1], p. 43–45) et reproduit avec sa solution dans ([2], p. 100–104).

$\theta \mapsto |\sin\theta| \times \ell/2$, c'est-à-dire

$$2\int_0^{2\pi} |\sin\theta| \times \frac{\ell}{2}\,\frac{\mathrm{d}\theta}{2\pi \times d} = \int_0^{\pi} \sin\theta \times \ell\,\frac{\mathrm{d}\theta}{\pi \times d} = \frac{2\ell}{\pi \times d}\,.$$

EXERCICE 1 (HORS SÉANCE DE TP). — Que vaut cette probabilité si $\ell > d$?

⚄ Nous allons mettre en œuvre l'expérience de Buffon pour obtenir une estimation de π.

```
// L'aiguille de Buffon

clear; mode(0);

N = 100;// taille des \'echantillons

// param\'etres ``physiques''
d = 2; // largeur des lattes
ell = 1; // longueur des aiguilles
ptheo = 2*ell/(%pi*d);

y = d*grand(N, 1, "def");// ordonn\'ees du centre des aiguilles
theta = 2*%pi*grand(N, 1, "def");// angles des aiguilles

// vecteur indicateur et proportion des croisements
flag = y+ell/2*abs(sin(theta)) >= d | y-ell/2*abs(sin(theta)) <= 0;
pobs = sum(bool2s(flag))/N;

// R\'esultats
mprintf("Probabilite theorique : %f\n", ptheo);
mprintf("Proportion observee : %f\n", pobs);
mprintf("Estimation de pi : %f\n", 2*ell/d/pobs);
```

Remarque. — Noter que `y` et `theta` sont des vecteurs. Ainsi, dans la ligne définissant la quantité `flag`, les opérations se font alors coordonnée par coordonnée, et même les opérations logiques ! En regardant bien, on voit que `flag` est le résultat de deux comparaisons connectées par un « ou » codé par la barre verticale, le résultat est donc un booléen, ou plutôt, un vecteur de booléens. Pour calculer `pobs` nous avons d'abord converti les coordonnées de `flag` en une suite de 0 et de 1 avant d'effectuer la somme via la commande `bool2s`.

⚄ Le résultat est-il satisfaisant ? Faire varier n, d et ℓ et observer.

⚄ Comme on le sait, une estimation ponctuelle seule ne signifie pas grand'chose. Chaque estimation est différente de la valeur exacte de π, elles sont même rationnelles. Nous passons donc à une estimation par intervalles :

```
alpha = 0.05;
z = cdfnor("X", 0, 1, 1-alpha/2, alpha/2);
mprintf("Intervalle de confiance de ptheo = %f\n", 2*ell/(%pi*d));
mprintf("borne inferieure = %f\n", pobs-z*sqrt(pobs*(1-pobs)/N));
mprintf("borne superieure = %f\n", pobs+z*sqrt(pobs*(1-pobs)/N));
```

⚄ En déduire un intervalle de confiance pour π.

⚄ Nous allons maintenant étudier l'évolution de l'estimation au fil des essais. Le calcul des intervalles de confiance se fera par une méthode moins sommaire que celle qui précède (méthode dite de Laplace) : la méthode Clopper–Pearson. Nous ne détaillons pas la méthode en question, elle est simplement mise en place dans le fichier `CLI.sce`.

```
// Intervalles de confiance de la probabilit\'e th\'eorique

exec CLI.sce;// fichier re\c cu par mail
// \`a placer dans son r\'epertoire de travail

k = 0; p = zeros(N,1); pmin = zeros(N,1); pmax = zeros(N,1);
for n = 1:N;
    if flag(n) then k = k+1; end
        p(n) = k/n;
        [pmin(n), pmax(n)] = proportionCLI(k, n, alpha);
end

scf(0); clf(); plot2d([1:N]', [p, pmin, pmax]);
xset("color", 5); xpoly([1; N], [ptheo; ptheo]);
// On marque et compte les intervalles incorrects
proportion = 0;
for n = 1:N;
    if pmin(n) > ptheo | pmax(n) < ptheo then proportion = proportion+1;
        xpoly([n; n], [pmin(n); pmax(n)]);
    end
end
proportion = proportion/N;
mprintf("Proportion d''estimations par intervalles incorrectes : %f\n", ...
     proportion);
xset("color", 0);
legend(["proportion observee"; "borne inferieure"; ...
    "borne superieure"; "probabilite theorique"], 1);
xtitle("l''aiguille de Buffon");
```

L'appartenance ou non aux différents intervalles de confiance de la probabilité théorique est-elle une illustration directe de la notion d'intervalle de confiance ? Répéter la simulation.

Exercice 2 (hors séance de TP). — Revenons au jeu du franc carreau. En notant r le rayon de la pièce et d la longueur des côtés des carreaux, après une modélisation convenable, déterminer la probabilité de succès.

2. Méthode de Monte Carlo pour le calcul approché d'intégrales

Soit $D = [a_1, b_1] \times \cdots \times [a_d, b_d] \subset \mathbb{R}^d$ un pavé fermé borné et $f : D \to \mathbb{R}$ une fonction intégrable (par rapport à la mesure de Lebesgue). On souhaite calculer

$$I = \int_D f(x_1, \ldots, x_d)\,\mathrm{d}x_1 \ldots \mathrm{d}x_n = \int_D f(x)\,\mathrm{d}x$$

au moins de manière approchée. Pour ce faire, on considère une suite $(X_n)_{n\geqslant 1}$ de variables aléatoires indépendantes identiquement distribuées de loi uniforme sur D, et on pose

$$I_n = \frac{\mathrm{Vol}(D)}{n} \sum_{k=1}^{n} f(X_k), \qquad n \geqslant 1.$$

Approximation : estimation ponctuelle. Montrer à l'aide de la loi forte des grands nombres que la suite de variables aléatoires $(I_n)_{n\geqslant 1}$ converge presque sûrement vers I et que I_n est un estimateur sans biais de I.

⚄ *Contrôle de l'erreur : estimation par intervalle.* On suppose $f : D \to \mathbb{R}$ de carré intégrable. Calculer la variance de I_n, $n \geqslant 1$. En donner un estimateur. En déduire un intervalle de confiance asymptotique de I.

Une première intégrale. — Soient $D = [0,1]$ et $f(x) = \sqrt{1-x^2}$ pour $x \in [0,1]$. La valeur de I saute aux yeux, pourtant mettre en œuvre le calcul approché par la méthode proposée. En s'inspirant de ce qui a été fait dans le cas de l'aiguille de Buffon, tracer les estimations successives avec les bornes des intervalles de confiance correspondants.

Une seconde intégrale. — Soient $D = \{(x,y) \in [0,1]^2 : x^2+y^2 \leqslant 1\}$ et $f(x,y) = \sqrt{1-x^2-y^2}$ pour $(x,y) \in D$.

⚄ Déterminer la valeur exacte de I.

Le domaine D n'est pas un pavé mais il est néanmoins borné et en remplaçant f par $f \times \mathbb{1}_D$ et D par $[0,1]^2$, on se retrouve dans le cadre décrit en début de section.

⚄ Mettre en œuvre le calcul approché par la méthode proposée. En s'inspirant de ce qui a été fait dans le cas de l'aiguille de Buffon, tracer les estimations successives avec les bornes des intervalles correspondants.

Remarque. — Généralement, on ne connaît ni l'intégrale de f ni l'intégrale de f^2. Les variances sont donc assez mal contrôlées dans cette méthode qui est en quelque sorte une randomisation d'intégration de Riemann (échantillonnage du domaine d'intégration). Avec cette même optique, des méthodes plus avancées existent (Vegas, MISER) tenant mieux compte du problème de la variance.

3. **Méthode de Monte Carlo pour le calcul approché d'un volume**

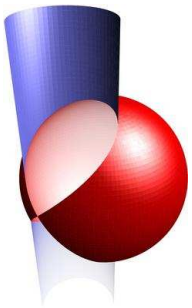
⚄ Nous allons estimer le volume de la boule unité B de $\mathbb{R}^3$ de la manière suivante : si $(X_1, \ldots, X_n)$ est une suite de variables aléatoires indépendantes identiquement distribuées de loi uniforme sur $[-1,1]^3$, alors $(\mathbb{1}_{\{X_1 \in B\}}, \ldots, \mathbb{1}_{\{X_n \in B\}})$ est une suite de variables aléatoires indépendantes identiquement distribuées de loi de Bernoulli de paramètre

$$p = \frac{\mathrm{Vol}(B)}{\mathrm{Vol}([-1,1]^3)} = \frac{4\pi/3}{8} = \frac{\pi}{6}\,.$$

```
// Calcul approch\'e du volume de la boule unit\'e

clear; mode(0);
exec CLI.sce;
n = 100;
X = 2*grand(n,3,"def")-1;
flag = X(:,1).^2+X(:,2).^2+X(:,3).^2 <= 1;
k = sum(bool2s(flag));
alpha=0.05;
mprintf("Volume exact de la boule unite : %f\n", 4*%pi/3);
mprintf("Volume approche de la boule unite : %f\n", 8*k/n);
[pmin, pmax] = proportionCLI(k, n, alpha);
mprintf("intervalle de confiance : [%f, %f]\n", 8*pmin, 8*pmax);
```

On pourrait bien sûr regarder l'évolution progressive de l'estimation…

⚄ Le « temple de Viviani » est l'intersection de la boule unité et du cylindre plein passant par $(1/2, 0, 0)$, d'axe $z'Oz$ et de rayon $1/2$. Adapter ce qui précède à ce nouveau volume qui vaut exactement $\frac{2}{3}(\pi - 4/3)$. (À calculer soi-même !)



```
theta = linspace(0, 2*%pi, 100);
phi = linspace(-%pi/2, %pi/2, 50);
z = linspace(-2,2,100);
scf(0); clf();
plot3d2(cos(phi)'*cos(theta), cos(phi)'*sin(theta), sin(phi)'*ones(theta),...
    flag=[5, 3, 0]);
plot3d2(0.5*ones(z)'*(cos(theta)+1), 0.5*ones(z)'*sin(theta), z'*ones(theta),...
    flag=[2, 3, 0], ebox=[-1,1,-1,1,-2,2]);
//a = gca(); a.isoview="on"; a.cube_scaling="on";
title("Temple de Viviani");
```

Références

[1] Buffon (G.), « Editor's note concerning a lecture given 1733 by Mr. Le Clerc de Buffon to the Royal Academy of Sciences in Paris », Histoire de l'Acad. Roy. des Sci., p. 43–45, 1733.

[2] Buffon (G.), *Essai d'arithmétique morale*, Histoire naturelle, générale et particulière, Supplément 4, p. 46–123, 1777.

Université de Poitiers
Département de Mathématiques

TRAVAUX PRATIQUES. — FEUILLE N° 1, INTERVALLES DE CONFIANCE D'UNE PROPORTION

Une part non négligeable des simulations proposées consistent en une estimation par intervalles d'une proportion, ou d'une probabilité, qu'on note ici $p_{\text{théo}} \in [0,1]$. Nous disposons d'un échantillon $(X_1, \dots, X_n)$ de variables aléatoires indépendantes identiquement distribuées de loi $\mathcal{B}(1, p_{\text{théo}})$ — qui ont pour espérance $p_{\text{théo}}$ et pour variance $\sigma^2 = p_{\text{théo}}(1 - p_{\text{théo}})$ —, et

$$\bar{X}_n = \frac{1}{n}(X_1 + \cdots + X_n)$$

est un estimateur fortement consistant et sans biais de $p_{\text{théo}}$. Pour la variance échantillon (non corrigée), nous avons

$$\begin{aligned} nS_n^2 &= \sum_{k=1}^{n} \big(X_k - \bar{X}_n\big)^2 = \sum_{k=1}^{n} \Big(X_k^2 - 2\bar{X}_n X_k + \big(\bar{X}_n\big)^2\Big) = \sum_{k=1}^{n} \Big(X_k - 2\bar{X}_n X_k + \big(\bar{X}_n\big)^2\Big) \\ &= \sum_{k=1}^{n} X_k - 2\bar{X}_n \sum_{k=1}^{n} X_k + \sum_{k=1}^{n} \big(\bar{X}_n\big)^2 = n\bar{X}_n - 2n\bar{X}_n\bar{X}_n + n\big(\bar{X}_n\big)^2 \\ &= n\bar{X}_n\big(1 - \bar{X}_n\big) \end{aligned}$$

car pour tout k, $X_k^2 = X_k$ puisque ce sont des variables à valeurs dans $\{0,1\}$. On retient que

$$\bar{X}_n\big(1 - \bar{X}_n\big)$$

est un estimateur fortement consistant (mais biaisé) de la variance $\sigma^2 = p_{\text{théo}}(1 - p_{\text{théo}})$. Nous supposerons $\sigma^2 > 0$, c'est-à-dire $p_{\text{théo}} \in \,]0,1[$, les cas extrêmes nécessitant une approche différente.

Approche rapide. — Le théorème central limite affirme que

$$\frac{\bar{X}_n - p_{\text{théo}}}{\sqrt{p_{\text{théo}}(1 - p_{\text{théo}})/n}}$$

est de loi proche de la loi normale $\mathcal{N}(0,1)$ lorsque n est grand. On a en particulier

$$\mathbb{P}\left\{-z_{1-\alpha/2} \leqslant \frac{\bar{X}_n - p_{\text{théo}}}{\sqrt{p_{\text{théo}}(1 - p_{\text{théo}})/n}} \leqslant z_{1-\alpha/2}\right\} \approx 1 - \alpha \qquad (E_\alpha)$$

où $z_{1-\alpha/2} = -z_{\alpha/2}$ est le quantile d'ordre $1 - \alpha/2$ de la loi normale $\mathcal{N}(0,1)$. Puisque $\bar{X}_n$ approche $p_{\text{théo}}$, on devrait avoir aussi

$$\mathbb{P}\left\{-z_{1-\alpha/2} \leqslant \frac{\bar{X}_n - p_{\text{théo}}}{\sqrt{\bar{X}_n(1 - \bar{X}_n)/n}} \leqslant z_{1-\alpha/2}\right\} \approx 1 - \alpha$$

ce qui s'écrit

$$\mathbb{P}\left\{\bar{X}_n - z_{1-\alpha/2}\sqrt{\frac{\bar{X}_n(1 - \bar{X}_n)}{n}} \leqslant p_{\text{théo}} \leqslant \bar{X}_n + z_{1-\alpha/2}\sqrt{\frac{\bar{X}_n(1 - \bar{X}_n)}{n}}\right\} \approx 1 - \alpha$$

D'où l'intervalle de confiance asymptotique usuel d'une proportion :

$$\left[\bar{X}_n - z_{1-\alpha/2}\sqrt{\frac{\bar{X}_n(1-\bar{X}_n)}{n}}\,;\,\bar{X}_n + z_{1-\alpha/2}\sqrt{\frac{\bar{X}_n(1-\bar{X}_n)}{n}}\right] \qquad (L_\alpha)$$

Une approche plus prudente. — Revenons à (E_α) qui s'écrit aussi

$$\mathbb{P}\left\{\frac{\big(\bar{X}_n - p_{\text{théo}}\big)^2}{p_{\text{théo}}(1-p_{\text{théo}})/n} \leqslant z^2_{1-\alpha/2}\right\} \approx 1-\alpha$$

où encore

$$\mathbb{P}\Big\{\big(1+z^2_{1-\alpha/2}/n\big)p^2_{\text{théo}} - \big(2\bar{X}_n + z^2_{1-\alpha/2}/n\big)p_{\text{théo}} + \bar{X}_n^2 \leqslant 0\Big\} \approx 1-\alpha.$$

On voit apparaître un polynôme du second degré en la variable $p_{\text{théo}}$ qui est de coefficient dominant strictement positif. Ainsi, il est négatif lorsque $p_{\text{théo}}$ est entre ses racines :

$$P_{\min/\max} = \frac{\bar{X}_n + z^2_{1-\alpha/2}/2n \mp \sqrt{z^2_{1-\alpha/2}/n \times \big(z^2_{1-\alpha/2}/4n + \bar{X}_n(1-\bar{X}_n)\big)}}{1+z^2_{1-\alpha/2}/n} \qquad (W_\alpha)$$

qui définissent donc l'intervalle de confiance asymptotique cherché. Celui-ci est meilleur que le précédent si on se fonde uniquement sur l'approximation du théorème central limite. Il présente de plus le grand avantage de fournir des bornes qui sont toujours dans $[0,1]$ contrairement à ce qu'il se passe pour l'approche rapide. Cette méthode semble due à Wilson (1927), alors que la première était déjà connue de Laplace.

Le problème avec ces bornes d'intervalles de confiance est qu'on les retient difficilement par cœur. En Scilab, on peut définir

```
function [pmin, pmax] = CLI(k, n, alpha);
    // local z, t;
    z = cdfnor("X", 0, 1, 1-alpha/2, alpha/2); z = z.*z./n;
    t = sqrt(z.*(z./4+k./n.*(1-k./n)));
    pmin = (k./n+z./2-t)./(1+z); pmax = (k./n+z./2+t)./(1+z);
endfunction
```

où `CLI` signifierait *Confidence Limits/Interval*, et où on aura essayé de vectorialiser le calcul.

Une approche exacte. — Les deux méthodes précédentes sont naturelles lorsqu'on se base sur le théorème central limite. Pourtant le choix d'une condition symétrique dans (E_α) reste arbitraire et d'autres conditions auraient donné lieu à d'autres formes d'intervalles de confiance. En recherchant un intervalle de confiance exact, c'est-à-dire des bornes $P_{\min/\max} = \phi_{\min/\max}(X_1,\dots,X_n)$ telles que $\mathbb{P}\{P_{\min} \leqslant p_{\text{théo}} \leqslant P_{\max}\} = 1-\alpha$, l'arbitraire du choix de l'intervalle est peut-être plus flagrant. La détermination que nous présentons est celle de Clopper–Pearson.

Nous ne connaissons pas $p_{\text{théo}}$ mais observons k réussites sur n essais.

La probabilité qu'une variable aléatoire de loi $\mathcal{B}(n,p)$ prenne des valeurs aussi grandes que k est

$$P_{n,k}(p) = \sum_{\ell=k}^{n} C_n^\ell\, p^\ell (1-p)^{n-\ell}.$$

Lorsque $k \geqslant 1$, cette fonction polynomiale, positive sur $[0,1]$, est nulle en $p=0$ et on a presque immédiatement

$$P'_{n,k}(p) = \sum_{\ell=k}^{n} (\ell - np) C_n^\ell\, p^{\ell-1}(1-p)^{n-\ell-1}$$

et on constate que pour tous $p \in [0, k/n]$, $\ell \geqslant k$, on a $\ell - np \geqslant 0$, et qu'ainsi $P'_{n,p}$ est positive sur $[0, k/n]$ et donc que $P_{n,k}$ est croissante sur $[0, k/n]$, et même strictement croissante sur $]0, k/n[$, et on a grossièrement $P_{n,k}(k/n) \approx 1/2$. Pour α assez petit, il existe alors un unique $p_{\min} \in [0, k/n]$ tel que $P_{n,k}(p_{\min}) = \alpha/2$.

Le choix de la borne inférieure de l'intervalle de confiance se justifie ainsi : puisque nous avons observé k, la probabilité d'observer des valeurs aussi grandes que k doit être important, c'est-à-dire $P_{n,k}(p_{\text{théo}}) \geqslant \alpha/2$, ou encore $p_{\text{théo}} \geqslant p_{\min}$. La borne inférieure $p_{\min}$ est bien une fonction des observations $k = x_1 + \cdots + x_n$ (et aussi de la taille n de l'échantillon et du seuil α).

Pour la borne supérieure, on procède de même, ou bien en considérant les probabilités d'observer des valeurs aussi petites que k lorsque $k < n$, et la fonction polynomiale

$$Q_{n,k}(p) = \sum_{\ell=0}^{k} C_n^\ell \, p^\ell (1-p)^{n-\ell},$$

ou bien en passant formellement au complémentaire succès/échec. En résumé :

$$\begin{cases} P_{n,k}(p_{\min}) = \alpha/2 & \text{avec } p_{\min} \in [0, k/n], \\ Q_{n,k}(p_{\max}) = \alpha/2 & \text{avec } p_{\max} \in [k/n, 1]. \end{cases} \qquad (CP_\alpha)$$

Le problème de détermination des bornes est numérique : il faut inverser un polynôme dont les valeurs sont déjà lourdes à calculer. Avec le programme Scilab suivant, nous procédons par dichotomie.

```
accuracy = 1E-6;

function [pmin, pmax] = CLI(k, n, alpha);
    if n <= 92 then
        // Clopper-Pearson
        // local p, lb, ub;
        lb = 0; ub = k/n;
        while ub-lb > accuracy;
            p = .5*(lb+ub);
            if 1-cdfbin("PQ", k-1, n, p, 1-p) < alpha/2 then
        lb = p; else ub = p; end
        end
        pmin = 0.5*(lb+ub);
        lb = k/n; ub = 1;
        while ub-lb > accuracy;
            p = 0.5*(lb+ub);
            if cdfbin("PQ", k, n, p, 1-p) < alpha/2 then
        ub = p; else lb = p; end
        end
        pmax = 0.5*(lb+ub);
    else
        // Wilson
        // local p, z, t;
        p = k/n;
        z = cdfnor("X", 0, 1, 1-alpha/2, alpha/2); z = z*z/n;
        t = sqrt(z*(z/4+p*(1-p)));
        pmin = (p+z/2-t)/(1+z); pmax = (p+z/2+t)/(1+z);
    end
endfunction
```

La valeur du paramètre `accuracy` donnée ici semble suffisante, tout comme le recours à un calcul asymptotique pour de grandes valeurs de n. (Le « 92 » est pour les utilisateurs de Scilab purement anecdotique, il pourrait être remplacé par 100, voire peut-être par 1000).

Remarque. — Si $k = 0$ ou $k = n$, le calcul d'intervalle de confiance bilatéral est presque dépourvu de sens ; on regarde un intervalle de confiance unilatéral. Dans le cas exact, lorsque $k = n$ (resp. $k = 0$) on peut alors reprendre la même détermination de la borne inférieure (resp. supérieure) en remplaçant $\alpha/2$ par α, ce qui n'est pas fait dans ce programme ; la borne supérieure (resp. inférieure) est alors 1 (resp. 0).

Université de Poitiers
Département de Mathématiques

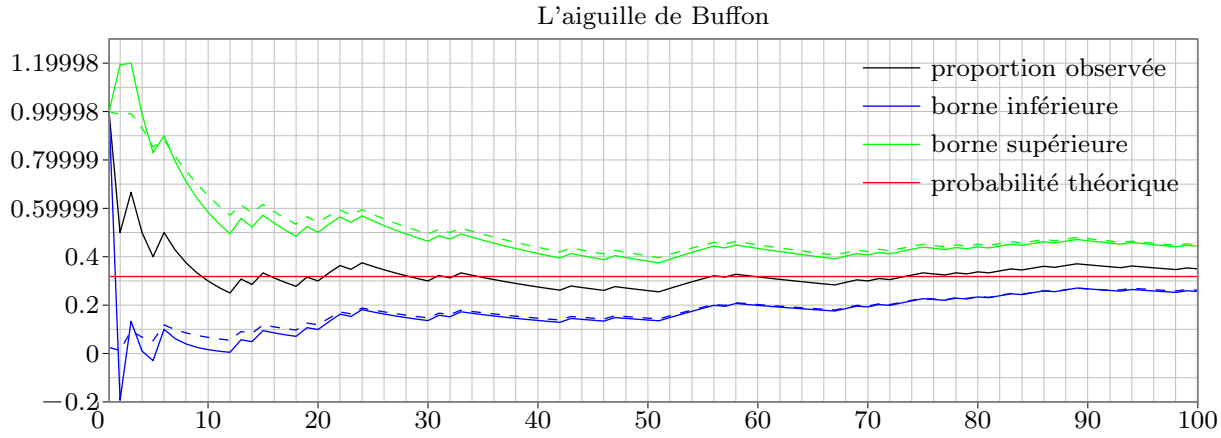
TRAVAUX PRATIQUES. — FEUILLE N° 1, ÉLÉMENTS D'EXPLICATIONS

1. L'aiguille de Buffon

L'estimateur de $p_{\text{théo}} = 2\ell/(\pi d)$ est bien sûr la proportion observée p_{obs} de croisements. Pour l'estimation par intervalle, on se donne pour seuil $\alpha = 0{,}05$ (ou pour niveau de confiance $1-\alpha = 0{,}95$), l'intervalle de confiance asymptotique usuel s'écrit alors

$$\left[p_{\text{obs}} - z_{1-\alpha/2}\sqrt{\frac{p_{\text{obs}}(1-p_{\text{obs}})}{n}} \,,\, p_{\text{obs}} + z_{1-\alpha/2}\sqrt{\frac{p_{\text{obs}}(1-p_{\text{obs}})}{n}} \right]$$

où n est la taille de l'échantillon et $z_{1-\alpha/2} \approx 1{,}96$ est le quantile d'ordre $1-\alpha/2 = 0{,}975$ de la loi normale $\mathcal{N}(0,1)$. Si pour $n \geqslant 20$, ce type d'intervalles convient en général, pour n petit, il peut donner des résultats curieux, sa borne inférieure pouvant être négative et sa borne supérieure plus grande que 1. Voici le type de graphique qu'on obtient :



Les bornes en traits pleins correspondent aux intervalles de confiance asymptotiques usuels, celles en pointillés aux intervalles exacts (Clopper–Pearson) pour une taille d'échantillons inférieure à 92 (limitations de MetaPost…) et à la seconde forme d'intervalles asymptotiques (Wilson) sinon. On constate que pour n assez grand, les deux méthodes donnent des résultats quasi-identiques.

Pour la sortie texte (ne correspondant pas nécessairement à cette figure) :

```
1. L'aiguille de Buffon
-----------------------
Probabilite theorique : 0.31831
Proportion observee : 0.32
Estimation de pi : 3.12492
Intervalle de confiance de ptheo = 0.31831
borne inferieure = 0.22845
borne superieure = 0.41156
Intervalle de confiance de pi = 3.14159
borne inferieure = 2.42978
borne superieure = 4.37724
```

```
Proportion d'estimations par intervalles incorrectes : 0.01
```

L'intervalle de confiance de π s'obtient en considérant l'équivalence

$$p_{\min} \leqslant p_{\text{théo}} = \frac{2\ell}{\pi \times d} \leqslant p_{\max} \iff \frac{2\ell}{p_{\max} \times d} \leqslant \pi \leqslant \frac{2\ell}{p_{\min} \times d}$$

qui donne immédiatement les bornes cohérentes avec l'estimation de $p_{\text{théo}}$.

2. **Méthode de Monte Carlo pour le calcul approché d'intégrales**

⚄ *Approximation : estimation ponctuelle.* Puisque les variables aléatoires $(X_n)_{n\geqslant 1}$ sont indépendantes et identiquement distribuées, il en est de même de $(Y_n)_{n\geqslant 1} = (f(X_n))_{n\geqslant 1}$. De plus, par le théorème de transfert,

$$\mathbb{E}[|Y_n|] = \mathbb{E}[|f(X_n)|] = \int_D |f(x)|\, P_{X_n}(\mathrm{d}x) = \int_D |f(x)|\, \frac{\mathrm{d}x}{\mathrm{Vol}(D)} < \infty$$

puisque X_n est de loi uniforme sur D et que f y est intégrable pour cette mesure. D'après la loi forte des grands nombres, nous avons presque sûrement

$$\frac{I_n}{\mathrm{Vol}(D)} = \bar{Y}_n = \frac{1}{n}\sum_{k=1}^{n} Y_k = \frac{1}{n}\sum_{k=1}^{n} f(X_k) \longrightarrow \int_D f(x)\, \frac{\mathrm{d}x}{\mathrm{Vol}(D)} = \frac{I}{\mathrm{Vol}(D)} \qquad \text{quand } n \to \infty.$$

Donc, la suite $(I_n)_{n\geqslant 1}$ est un estimateur fortement consistant de I. Il est de plus sans biais car pour tout n, I_n est intégrable et $\mathbb{E}[I_n] = I$.

⚅ *Contrôle de l'erreur : estimation par intervalle.* Supposons f de carré intégrable. Nous avons

$$\begin{aligned}\sigma^2 = \mathrm{Var}(Y_n) = \mathrm{Var}\big(f(X_n)\big) &= \mathbb{E}\big[f(X_n)^2\big] - \mathbb{E}[f(X_n)]^2 \\ &= \int_D f(x)^2\, \frac{\mathrm{d}x}{\mathrm{Vol}(D)} - \left(\int_D f(x)\, \frac{\mathrm{d}x}{\mathrm{Vol}(D)}\right)^2\end{aligned}$$

dont la valeur n'est *a priori* pas accessible. Nous nous contentons de son estimateur habituel (biaisé)

$$S_n^2 = \frac{1}{n}\sum_{k=1}^{n} \big(Y_k - \bar{Y}_n\big)^2 = \frac{1}{n}\sum_{k=1}^{n} \big(f(X_k) - I_n/\mathrm{Vol}(D)\big)^2 = \frac{1}{n}\sum_{k=1}^{n} f(X_k)^2 - \big(I_n/\mathrm{Vol}(D)\big)^2.$$

Ce qui donne pour intervalle de confiance asymptotique au niveau de confiance $1-\alpha$ de $I/\mathrm{Vol}(D)$

$$\left[\frac{I_n}{\mathrm{Vol}(D)} \mp z_{1-\alpha/2}\sqrt{\frac{S_n^2}{n}}\right] = \frac{1}{\mathrm{Vol}(D)}\left[I_n \mp z_{1-\alpha/2}\sqrt{\frac{\mathrm{Vol}(D)^2\big(\sum_{k=1}^{n} f(X_k)^2\big)/n - I_n^2}{n}}\right],$$

où $z_{1-\alpha/2}$ est le quantile d'ordre $1-\alpha/2$ de la loi normale $\mathcal{N}(0,1)$. D'où finalement l'intervalle de confiance asymptotique de I

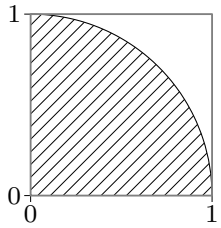
$$\left[I_n - z_{1-\alpha/2}\sqrt{\frac{\mathrm{Vol}(D)^2/n\sum_{k=1}^{n} f(X_k)^2 - I_n^2}{n}}\,,\; I_n + z_{1-\alpha/2}\sqrt{\frac{\mathrm{Vol}(D)^2\big(\sum_{k=1}^{n} f(X_k)^2\big)/n - I_n^2}{n}}\right],$$

où par la suite Vol(D) = 1 !

Remarque. — Pour ces intervalles de confiance, nous ne jugeons pas utile de considérer l'estimateur non biaisé de la variance, le facteur $n/(n-1)$ étant embarassant et inessentiel.

Une première intégrale. — ⚄ Nous considérons

$$f : [0,1] \longrightarrow [0,1] \quad x \longmapsto \sqrt{1-x^2} \quad \text{et} \quad I = \int_0^1 f(x)\,\mathrm{d}x = \int_0^1 \sqrt{1-x^2}\,\mathrm{d}x = \pi/4$$

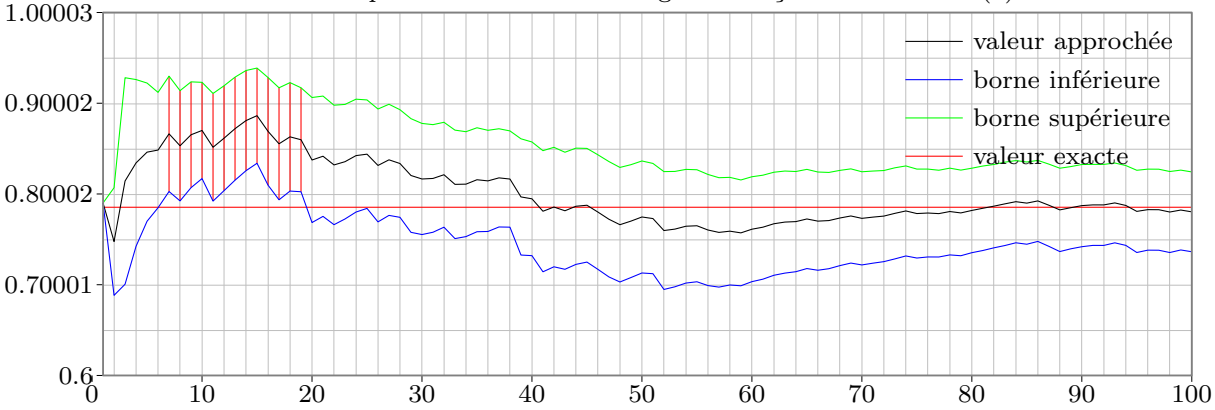


nombre qui sera désigné par `I` dans les programmes.

⚄ Pour la mise en œuvre en Scilab :

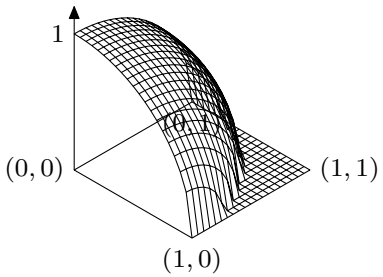
```
mprintf("\n2. Integration a la Monte Carlo (1)\n");

N = 100;// taille des \'echantillons

X = grand(N, 1, "def");
Y = sqrt(1-X.^2);
n = [1:N]';
In = cumsum(Y)./n;
Delta = 1.96*sqrt((cumsum(Y.^2)./n-In.^2)./n);
Imin = In-Delta; Imax = In+Delta;
I = %pi/4;

scf(1); clf();
plot2d(n, [In, Imin, Imax]);
xset("color", 5);
xpoly([1; N], [I; I]);
// On marque et compte les intervalles incorrects
proportion = 0;
for k = 1:N;
    if Imin(k) > I | Imax(k) < I then
        proportion = proportion+1;
        xpoly([k; k], [Imin(k); Imax(k)]);
    end
end
proportion = proportion/N;
xset("color", 0);
xtitle("Integration a la Monte Carlo (1)");
legend(["integrale approchee"; "borne inferieure"; ...
    "borne superieure"; "valeur exacte"]);
mprintf("Proportion d''estimations par intervalles incorrectes : %f\n", ...
    proportion);
```

2. Une première méthode d'intégration façon Monte Carlo (1)



Une seconde intégrale. — Nous considérons

$$f : [0,1]^2 \longrightarrow [0,1]$$
$$(x,y) \longmapsto \mathbb{1}_{\{x^2+y^2\leqslant 1\}}\sqrt{1-x^2-y^2}$$



et on a par changement de variables polaire, ou en remarquant que le volume est cherché est le huitième du volume de la boule unité,

$$I = \int_0^1\int_0^1 f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_0^1\int_0^{\pi/2} \sqrt{1-r^2}\times r\ \mathrm{d}\theta\,\mathrm{d}r$$
$$= \frac{\pi}{2}\int_0^1 r\sqrt{1-r^2}\,\mathrm{d}r = \frac{\pi}{2}\left[-\frac{1}{3}\left(1-r^2\right)^{3/2}\right]_0^1 = \frac{\pi}{6}$$

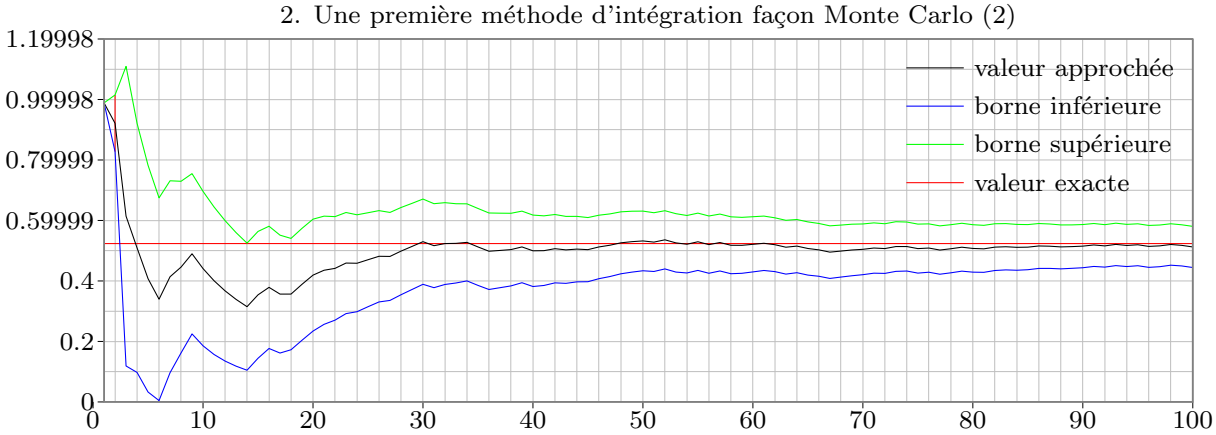
nombre qui sera désigné par `I` dans les programmes.

Pour la mise en œuvre en Scilab :

```
mprintf("\n2. Integration a la Monte Carlo (2)\n");

X = grand(N, 2, "def");
Y = sqrt(max(1-X(:, 1).^2-X(:, 2).^2, 0));
n = [1:N]';
In = cumsum(Y)./n;
Delta = 1.96*sqrt((cumsum(Y.^2)./n-In.^2)./n);
Imin = In-Delta; Imax = In+Delta;
I = %pi/6;// 4/3\pi\times1/8

scf(2); clf();
plot2d(n, [In, Imin, Imax]);
xset("color", 5);
xpoly([1; N], [I; I]);
// On marque et compte les intervalles incorrects
proportion = 0;
for k = 1:N;
    if Imin(k) > I | Imax(k) < I then
        proportion = proportion+1;
        xpoly([k; k], [Imin(k); Imax(k)]);
```

```
    end
end
proportion = proportion/N;
xset("color", 0);
xtitle("Integration a la Monte Carlo (2)");
legend(["integrale approchee"; "borne inferieure"; ...
    "borne superieure"; "valeur exacte"]);
mprintf("Proportion d''estimations par intervalles incorrectes : %f\n", ...
    proportion);
```

2. Une première méthode d'intégration façon Monte Carlo (2)



3. Méthode de Monte Carlo pour le calcul approché d'un volume

```
// Calcul approch\'e du volume de fenetre de Viviani

clear; mode(0);
exec CLI.sce;
n = 1000;
X = grand(n, 3, "def");
X(:,2) = X(:, 2)-1/2;
flag = X(:,1).^2+X(:,2).^2+X(:,3).^2 <= 1 & (X(:,1)-1/2).^2+X(:,2).^2 <= 1/4;
k = sum(bool2s(flag));
alpha=0.05;
mprintf("Volume exact de l''intersection : %f\n", 2/3*(%pi-4/3));
mprintf("Volume approche de l''intersection : %f\n", 2*k/n);
[pmin, pmax] = proportionCLI(k, n, alpha);
mprintf("intervalle de confiance : [%f, %f]\n", 2*pmin, 2*pmax);
clf();
param3d1(X(flag,1), X(flag,2), list(X(flag,3), -1), ...
    flag = [1,3], ebox=[-1,1,-1,1,-1,1]);
```

TRAVAUX PRATIQUES. — FEUILLE N° 2

Comme certains l'écrivent, « tout programme informatique faisant appel à un générateur de nombres pseudo-aléatoires simule une chaîne de Markov ».

Plus précisément, il a été vu en travaux dirigés que si $(E, \mathcal{E})$ est un espace d'états, $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé et $(F, \mathcal{F})$ un espace mesurable,

$$X_0 : (\Omega, \mathcal{A}, \mathbb{P}) \to (E, \mathcal{E}), \qquad U_n : (\Omega, \mathcal{A}, \mathbb{P}) \to (F, \mathcal{F}), \quad n \geqslant 1,$$

sont des variables aléatoires indépendantes et si

$$f_n : (E \times F, \mathcal{E} \otimes \mathcal{F}) \to (E, \mathcal{E}), \qquad n \geqslant 1,$$

sont des applications mesurables, alors le processus à temps discret $X = (X_n)_{n\geqslant 0}$ défini par

$$X_{n+1} = f_{n+1}(X_n, U_{n+1}), \qquad n \geqslant 0,$$

est markovien dans sa filtration naturelle mais aussi dans la filtration engendrée par X_0 et $(U_n)_{n\geqslant 1}$. Lorsque les variables aléatoires $(U_n)_{n\geqslant 1}$ ont même loi et lorsque $f_n = f$ pour tout $n \geqslant 1$, le processus est homogène et son noyau de transition peut être exprimé en fonction de la loi et de l'application f : $P(x, \mathrm{d}y) = \mathbb{P}\{f(x, U) \in \mathrm{d}y\}$. Il est possible d'énoncer une réciproque à cette propriété.

Le propos de ces travaux pratiques ne sera donc pas de faire la théorie de la simulation de chaînes de Markov puisque c'est déjà fait. Il sera simplement de simuler quelques systèmes simples et de comparer le comportement avec les prédictions théoriques.

1. **Généralités**

Puisque tout programme utilisant un générateur aléatoire donne lieu à une chaîne de Markov, les outils fondamentaux sont les générateurs de nombres pseudo-aléatoires associés à la commande `grand()`.

L'évolution d'une chaîne peut être décrite par des structures conditionnelles, des opérations arithmétiques, etc. ou par la « simple » donnée d'une matrice de transition.

Dans cette dernière situation, rappelons qu'on considère un espace d'états fini dont les états peuvent être identifiés avec les entiers $\{1, \ldots, k\}$. S'il est besoin de leur donner un nom, on pourra utiliser un vecteur de labels. Par exemple,

```
E = ["++"; "+-"; "-+"; "--"];
```
et ainsi
```
E(3) = "-+",
```

ou encore

```
E = [0,0,0; 1,0,0; 0,1,0; 0,0,1; 1,1,0; 1,0,1; 0,1,1; 1,1,1];
```

et ainsi `E(7,:) = 0 1 1` les coordonnées du septième sommet d'un cube. Coder l'espace des états comme un vecteur colonne ou un vecteur ligne n'a pas de sens particulier, il n'en est cependant rien des mesures, des fonctions et donc des matrices de transition. Il faut suivre une convention, celle de la théorie mathématique : *avec* Scilab*, on codera les mesures par des vecteurs lignes, les fonctions par des vecteurs colonnes.* De sorte que si `P` code une matrice de transition en Scilab, le nombre `P(x,y)` est la probabilité de transiter de x à y, avec x, $y \in \{1, \ldots, k\}$.

Étant donnée une matrice `P`, la première question à se poser est de savoir s'il s'agit d'une matrice de transition. Scilab n'a pas de fonction prédéfinie pour effectuer cette vérification. On peut programmer

```
// 1. G\'en\'eralit\'es

clear(); mode(0); usecanvas(%f);

function flag = ismarkov(P);
    // local s;
    flag = size(P,1) == size(P,2);
    if flag then
        for x = 1:size(P,1);
            s = 0;
            for y = 1:size(P,2);// \'egal \`a size(P,1)
                flag = flag & P(x,y) >= 0; s = s+P(x,y);
            end
            flag = flag & s == 1;
        end
    end
    if flag then mprintf("C''est une matrice markovienne.\n");
    else mprintf("Ce n''est pas une matrice markovienne.\n");
    end
endfunction
```

Ce morceau de programme n'est guère profond mais peut aider à détecter des erreurs de saisie.

Taper la fonction Scilab précédente. Saisir la matrice

$$P = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Vérifier qu'il s'agit d'une matrice de transition, d'une matrice markovienne :

```
ismarkov(P)
```

Sa matrice transposée `P'` est-elle markovienne ? Le vérifier avec Scilab.

Cette matrice particulière a été vue en travaux dirigés. Il s'agissait de déterminer ses classes communicantes. Avec un peu d'habitude, elles se lisent sur le diagramme de transition correspondant, avec plus d'habitude, directement sur la matrice de transition.

Évidemment Scilab permet de calculer numériquement les puissances explicites $(P^n)_{n\geqslant 0}$ d'une matrice de transition et d'ainsi détecter d'éventuelles convergences, ou de regarder la suite des lois $(\mu P^n)_{n\geqslant 0}$ de la chaîne avec pour loi initiale μ.

Exécuter les lignes suivantes :

```
mu = ones(1, size(P,1))/size(P,1);// loi uniforme sur E, mu = (1,1,1,1,1,1)/6;
mu*P
mu*P^2
mu*P^3
mu*P^4
mu*P^100
```

La suite converge-t-elle ? si oui, quelle est sa limite ? Dépend-t-elle du choix de μ ? Le voir directement sur les puissances de P.

Il est tout de même plus amusant de simuler des trajectoires de la chaîne. Étant donné X_n, nous tirons au hasard X_{n+1} selon la loi $P(X_n, .)$, ce que nous savons faire :

```
function y = markovstep(x, P);
    // local p;
    p = grand(1, 1, "def"); y = 1;
    while p > P(x,y); p = p-P(x,y); y = y+1; end
endfunction

T = 100; x = 1; // valeur initiale
for i = 1:T; x = [x; markovstep(x(i), P)]; end
clf(); plot(x, "o-");
```

⚄ Exécuter cette portion de programme. La trajectoire obtenue est-t-elle illustrative de ce que nous savons sur la dynamique de la chaîne ?

Noter que le vecteur obtenu est indexé de 1 à $n + 1 = 101$. Que ce soit un vecteur ligne ou un vecteur colonne n'est pas, nous semble-t-il, important pour ce qui est du formalisme markovien. Ceci peut être fait plus efficacement avec Scilab en utilisant :

`grand(`$\langle n \rangle$`, "markov",` $\langle P \rangle$`,` $\langle x_0 \rangle$`)`

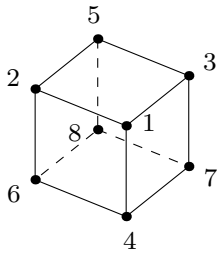
Cette commande renvoie un vecteur ligne de taille n et de première coordonnée x_0. Intégrer ce type de calculs dans une primitive Scilab est pertinent pour des questions de rapidité d'exécution. Ce genre de simulations intervenant dans des problèmes courants d'optimisation, il est normal qu'il soit lui-même optimisé... Cependant, il est aussi fréquent d'arrêter une simulation à un temps d'atteinte, et pour cela, on doit faire la simulation pas à pas.

⚄ Reprendre la simulation avec l'option `"markov"` de `grand()`.

```
y = grand(T+1, "markov", P, x(1))';// Z, c'est au d\'epart un vecteur ligne
clf(); plot([0:T]', [x,y], "o-");
```

2. **Marche aléatoire sur le cube**

Nous reprenons l'exercice sur la marche aléatoire sur le cube. Convenons de la numérotation des sommets :



⚄ Saisir la matrice de transition P correspondant à la chaîne étudiée :

```
// 2. Marche al\'eatoire sur le cube
// cardinal de l'espace d'\'etats k = 8;

P = [0,1,1,1,0,0,0,0; 1,0,0,0,1,1,0,0; 1,0,0,0,1,0,1,0; 1,0,0,0,0,1,1,0;
    0,1,1,0,0,0,0,1; 0,1,0,1,0,0,0,1; 0,0,1,1,0,0,0,1; 0,0,0,0,1,1,1,0]/3;
```

Vérifier que celle-ci est bien markovienne (qu'il n'y a pas eu d'erreur de saisie triviale), et qu'elle est symétrique.

Par simulation (en utilisant `markovstep` par exemple), la marche étant issue de $i = 1$, évaluer numériquement les quantités suivantes :

⚄ le temps moyen avant que la particule ne revienne en $i = 1$ ($\mathbb{E}^i[T^i_{\mathrm{r}}]$) ;

⚄ le nombre moyen de visites de $o = 8$ avant son premier retour en $i = 1$ ($\mathbb{E}^i[\sum_{n=0} T^i_{\mathrm{r}} \mathbb{1}_{\{X_n=o\}}]$) ;

⚄ le temps moyen avant la première visite de $o = 8$ ($\mathbb{E}^i[T^o]$).

Les moyennes pourront être observées sur $N = 100$, puis $N = 1000$, simulations de trajectoires qu'on arrêtera au temps d'atteinte ou au temps d'entrée correspondant.

⚄ Retrouver les moyennes $m(i, j) = \mathbb{E}^i[T^j_{\mathrm{r}}]$ pour $i = 1, \ldots, 8$, et $j = 8$.

Noter qu'arrêtant les trajectoires à des temps pseudo-aléatoires, l'usage de `grand()` pour générer de longues trajectoires (de taille fixe) est inadapté.

3. **Le contrôle des naissances**

L'exercice sur le « contrôle des naissances » est bien connu. Simuler ce qu'il peut se passer est facile :

```
clear; mode(0);

goal = "GGG";
wgoal = length(goal);
n = 10000;

function x = FoG();// Fille ou Gar\c con
   if grand(1, 1, "def") < 0.5 then x = "G"; else x = "F"; end
endfunction

Tbar = 0;

for i = 1:n;
  lastchildren = "";
  for j = 1:wgoal; lastchildren = lastchildren+FoG(); end
  T = wgoal;
  while lastchildren <> goal;
    lastchildren = part(lastchildren, 2:wgoal)+FoG();
    T = T+1;
  end
  Tbar = (i-1)/i*Tbar+T/i;// actualisation de la moyenne empirique
end

mprintf("Controle des naissances, atteindre ...%s, temps moyen observe %f\n", ...
     goal, Tbar);
```

⚄ Comprendre le programme ci-dessus.

⚄ Le mettre en œuvre pour estimer les temps moyens qui apparaissent dans les travaux dirigés ou contrôles.

⚄ Formaliser de manière markovienne le ou les problèmes considérés (l'espace d'états devant correspondre être cohérent avec le but à atteindre).

4. **L'urne d'Ehrenfest**

Nous nous donnons deux urnes contenant n boules en tout. Si au temps n la première urne contient contient x boules, on puise une boule de la première urne pour la mettre dans la seconde avec probabilité x/n, sinon, on puise une boule dans la seconde urne pour la mettre dans la première. La suite de variables aléatoires $(X_i)_{i\geqslant 0}$ des nombres successifs de boules dans la première urne est (sous des hypothèses raisonnables) une chaîne de Markov homogène de matrice de transition que nous n'écrirons pas. Les règles d'évolution, la chaîne, sont regroupés sous l'appellation « urne d'Ehrenfest ».

⚄ Définir une fonction `ehrenstep(`$\langle x\rangle$`, `$\langle n\rangle$`)` renvoyant aléatoirement l'état de la première urne après tirage.

⚄ Tracer quelques trajectoires $(x_i)_{i=0}^{T}$ pour une valeur initiale donnée et un temps terminal T raisonnable.

⚄ Tracer des diagrammes en bâtons d'échantillons de X_T et X_{T+1} pour un temps déterministe T assez grand devant n le nombre total de boules de l'urne et une taille d'échantillon N raisonnable.

⚄ Représenter une loi trajectorielle, c'est-à-dire la loi d'un échantillon observé consistant en une trajectoire $(x_i)_{i=0}^{N}$, par un diagramme en bâtons.

⚄ Commenter ces résultats au regard des résultats théoriques obtenus en travaux dirigés. On pourrait même comparer les lois observées avec les lois limites à l'aide d'un test du χ^2 (déjà pratiqué avec la procédure `dgoftest` et approfondi en Statistique inférentielle)...

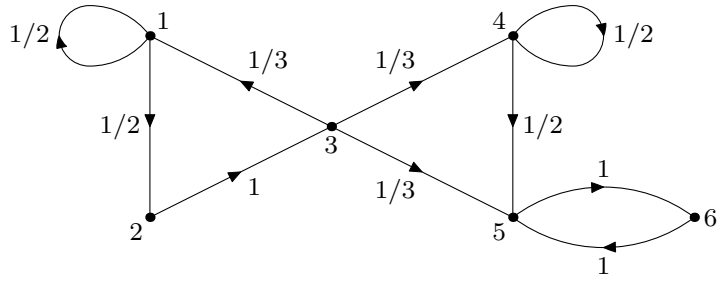
Remarque. — Les lois observées sont portées par $\{0, 1, \ldots, n-1, n\}$. On représentera ces lois par des vecteurs lignes $\langle loi\rangle$ de dimension $n+1$ dont chaque entrée contiendra le nombre d'observations correspondant convenablement normalisé. Les diagrammes en bâtons s'obtiennent alors avec `bar([0:n]',`$\langle loi\rangle$`)` ou encore `bar([0:n]',[`$\langle loi_1\rangle$`,`$\langle loi_2\rangle$`,`$\langle loi_3\rangle$`])`. L'usage de titres pour les graphiques (commande `xtitle`) ainsi que des légendes si besoin est (commande `legend` ou `legends`) est vivement conseillé.

Université de Poitiers
Département de Mathématiques

TRAVAUX PRATIQUES. — FEUILLE N$^{\text{O}}$ 2, ÉLÉMENTS D'EXPLICATIONS

1. **Généralités**

Rappelons le graphe de la matrice de transition P



Le seul commentaire qu'on peut faire est que l'instruction conditionnelle

```
if ismarkov(P) then mprintf("P est une matrice markovienne.\n");
else mprintf("P n''est pas une matrice markovienne.\n");
end
```

peut être remplacée par un simple

```
ismarkov(P)
```

la réponse de Scilab pouvant être « `T` » ou « `F` », ce qui est parfaitement explicite. On ne constate parmi les étudiants aucune erreur de saisie *involontaire* et `ismarkov(P')` donne « `F` », comme attendu.

Pour l'utilisation de `classmarkov()`, voici ce que ça donne avec P :

```
Eperm  =

   5.    6.    4.    1.    2.    3.
ans  =

   0.          1.    0.          0.          0.    0.
   1.          0.    0.          0.          0.    0.
   0.5         0.    0.5         0.          0.    0.
   0.          0.    0.          0.5         0.5   0.
   0.          0.    0.          0.          0.    1.
   0.3333333   0.    0.3333333   0.3333333   0.    0.
```

On devrait pouvoir lire sur cette matrice que la classe récurrente est $\{5, 6\}$ et que les classes transientes sont $\{4\}$ et $\{1, 2, 3\}$. Convaincant ? En fait, en regardant l'aide, on constate que la commande `classmarkov` donne beaucoup de détails sur la décompositon d'une matrice de transition. Ce peut être utile pour des matrices de grande taille, mais à un niveau pédagogique élémentaire, le plus difficile nous semble de comprendre l'aide.

D'autres fonctions sont proposées : `genmarkov` pour générer aléatoirement des matrices markoviennes ayant des caractéristiques données, et `eigenmarkov` pour déterminer les vecteurs propres (*eigenvectors* en allemand et en anglais), ou caractéristiques, de la matrice et sa transposée pour la valeur propre 1 (*eigenvalue* en allemand et en anglais). À retenir...

⚄ Pour le calcul de μP^n, on obtient

```
--> mu*P
 ans  =

    0.1388889    0.0833333    0.1666667    0.1388889    0.3055556    0.1666667
--> mu*P^2
 ans  =

    0.125    0.0694444    0.0833333    0.125    0.2916667    0.3055556
--> mu*P^3
 ans  =

    0.0902778    0.0625    0.0694444    0.0902778    0.3958333    0.2916667
--> mu*P^4
 ans  =

    0.0682870    0.0451389    0.0625    0.0682870    0.3599537    0.3958333
--> mu*P^100
 ans  =

    1.981D-12    1.276D-12    1.643D-12    1.981D-12    0.4666667    0.5333333
```

Nous connaissons les classes transientes $\{1,2,3\}$ et $\{4\}$, et la classe récurrente $\{5,6\}$. On sait donc que la masse totale va fuire les classes transientes pour se réfugier dans l'unique classe récurrente. C'est bien ce qu'on observe en partant de la mesure uniforme. On pourrait être tenté de croire qu'il va y avoir convergence vers la mesure uniforme sur $\{5,6\}$, c'est-à-dire vers $(0,0,0,0,1/2,1/2)\ldots$ Les étudiants s'étant exprimé n'étaient pas dupes. La classe $\{5,6\}$ est périodique de période 2, si bien que si on regarde μP^{101}, μP^{102}, on constate que les deux dernières coordonnées ne font qu'approximativement s'échanger l'une et l'autre. Leur proximité avec $1/2$ est presque fortuite. D'ailleurs, si on avait pris pour $\mu = \delta_{\{5\}}$ par exemple, la périodicité aurait été évidente.

Voici quelques puissances exactes de P :

$$P^2 = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad P^3 = \begin{pmatrix} \frac{7}{24} & \frac{1}{8} & \frac{1}{4} & \frac{1}{6} & \frac{1}{6} & 0 \\ \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} & \frac{1}{6} & \frac{1}{3} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} & \frac{1}{12} & \frac{5}{12} & \frac{1}{6} \\ 0 & 0 & 0 & \frac{1}{8} & \frac{5}{8} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$P^4 = \begin{pmatrix} \frac{11}{48} & \frac{7}{48} & \frac{1}{8} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{12} & \frac{1}{6} & \frac{1}{12} & \frac{5}{12} & \frac{1}{6} \\ \frac{7}{72} & \frac{1}{24} & \frac{1}{12} & \frac{7}{72} & \frac{19}{72} & \frac{5}{12} \\ 0 & 0 & 0 & \frac{1}{16} & \frac{5}{16} & \frac{5}{8} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad P^5 = \begin{pmatrix} \frac{5}{32} & \frac{11}{96} & \frac{7}{48} & \frac{1}{8} & \frac{7}{24} & \frac{1}{6} \\ \frac{7}{72} & \frac{1}{24} & \frac{1}{12} & \frac{7}{72} & \frac{19}{72} & \frac{5}{12} \\ \frac{11}{144} & \frac{7}{144} & \frac{1}{24} & \frac{11}{144} & \frac{71}{144} & \frac{19}{72} \\ 0 & 0 & 0 & \frac{1}{32} & \frac{21}{32} & \frac{5}{16} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

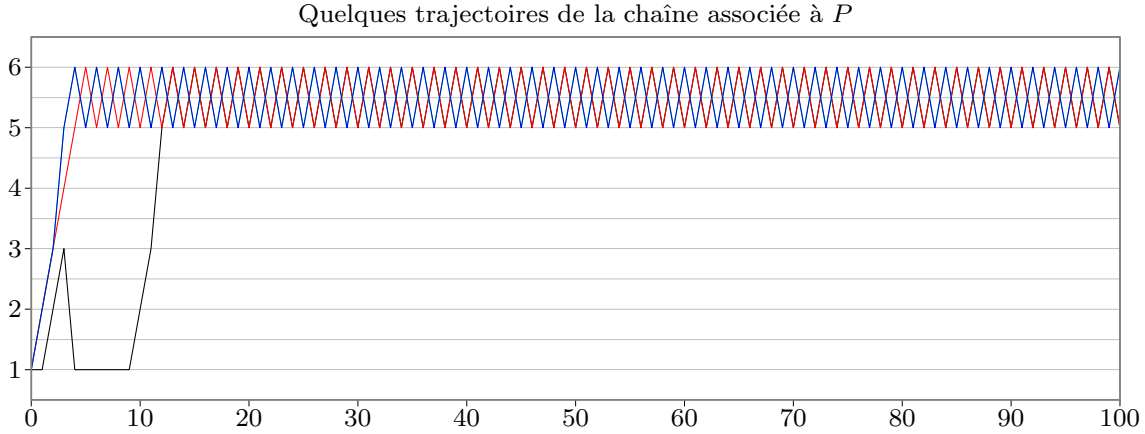
$$P^6 = \begin{pmatrix} \frac{73}{576} & \frac{5}{64} & \frac{11}{96} & \frac{1}{9} & \frac{5}{18} & \frac{7}{24} \\ \frac{11}{144} & \frac{7}{144} & \frac{1}{24} & \frac{11}{144} & \frac{71}{144} & \frac{19}{72} \\ \frac{5}{96} & \frac{11}{288} & \frac{7}{144} & \frac{5}{96} & \frac{91}{288} & \frac{71}{144} \\ 0 & 0 & 0 & \frac{1}{64} & \frac{21}{64} & \frac{21}{32} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad P^7 = \begin{pmatrix} \frac{13}{128} & \frac{73}{1152} & \frac{5}{64} & \frac{3}{32} & \frac{37}{96} & \frac{5}{18} \\ \frac{5}{96} & \frac{11}{288} & \frac{7}{144} & \frac{5}{96} & \frac{91}{288} & \frac{71}{144} \\ \frac{73}{1728} & \frac{5}{192} & \frac{11}{288} & \frac{73}{1728} & \frac{925}{1728} & \frac{91}{288} \\ 0 & 0 & 0 & \frac{1}{128} & \frac{85}{128} & \frac{21}{64} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

puis, de manière plus approximative,

$$P^{40} \approx \begin{pmatrix} 0 & 0 & 0 & 0 & 0.46658 & 0.53323 \\ 0 & 0 & 0 & 0 & 0.59991 & 0.39993 \\ 0 & 0 & 0 & 0 & 0.39993 & 0.59991 \\ 0 & 0 & 0 & 0 & 0.3333 & 0.66661 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad P^{41} \approx \begin{pmatrix} 0 & 0 & 0 & 0 & 0.53326 & 0.46658 \\ 0 & 0 & 0 & 0 & 0.39993 & 0.59991 \\ 0 & 0 & 0 & 0 & 0.59991 & 0.39993 \\ 0 & 0 & 0 & 0 & 0.66661 & 0.3333 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

etc. À la limite, on doit voir apparaître l'alternance de fractions simples (à établir proprement un jour), montrant la périodicité asymptotique.

Voici le type de trajectoires qu'on peut observer :



C'est si périodique que c'en est ennuyeux. L'utilisation de `grand()` avec l'option `"markov"` n'apporte rien de manifeste dans cette activité, mais on peut garder cette commande en mémoire.

2. Marche aléatoire sur le cube

La matrice de transition considérée se déduit facilement de la matrice d'adjacence du graphe puisque tous les sommets ont même valence : c'est cette même matrice d'adjacence multipliée par l'inverse de la valence :

$$P = \frac{1}{3}\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Cette matrice est donc symétrique.

Lorsque la valence n'est pas constante, la matrice de transition « naturellement » associée s'obtient en normalisant les lignes de la matrice d'adjacence par la somme de leurs coefficients. La matrice obtenue n'est plus nécessairement symétrique. Puisque nous discutons des matrices de transition qu'on peut associer à un graphe, si on note P la matrice de transition obtenue par normalisation de la matrice d'adjacence, pour $p \in [0,1]$, la matrice $P_p = p\,\mathrm{Id} + (1-p)P$ est encore une matrice de transition (les combinaisons barycentriques de matrices de transition sont des matrices de transition). Cette matrice de transition P_p autorise à stationner en chaque sommet ce qui est une condition suffisante d'apériodicité. Cette altération de la matrice de

transition, lorsque $p < 1$, a le bon goût de ne pas modifier les classes communiquantes (le voir directement sur la relation « mener à »), ni même de changer l'ensemble des mesures invariantes : $\lambda P = \lambda \iff (1-p)\lambda P = (1-p)\lambda \iff p\lambda + (1-p)\lambda P = p\lambda + (1-p)\lambda \iff \lambda P_p = \lambda$. Même si la matrice d'origine P n'est pas apériodique, P_p l'est, et la dynamique markovienne associée satisfait une propriété de convergence vers « un équilibre ». Dans le cas d'un graphe, l'irréductibilité de la matrice de transition équivaut à la connexité du graphe. Ainsi, si le graphe est connexe, la dynamique associée à P_p, pour $0 < p < 1$, converge vers l'équilibre, c'est-à-dire que les chaînes associées convergent vers l'unique mesure invariante π donnée par

$$\pi\{x\} = \frac{v\{x\}}{\sum_{y\in E} v\{y\}}, \qquad x \in E,$$

où $v\{x\}$ est la valence du sommet x, sommet qui peut être déjà relié à lui-même ou non.

Revenons au cube. Le graphe est connexe, donc la matrice de transition est irréductible, et comme nous sommes en dimension (ou cardinal) finie, la chaîne, ou la matrice est (positivement) récurrente et son unique mesure de probabilité invariante est la mesure uniforme $\pi\{x\} = 1/8$ pour tout x sommet du cube. D'ailleurs, nous avons :

Proposition. — *Soit P un noyau de transition sur un ensemble E au plus dénombrable. Si P est symétrique, $P(x,y) = P(y,x)$ pour tous x, $y \in E$, P est réversible par rapport à la mesure uniforme $(1)_{x\in E}$, mesure qui est donc invariante. Si de plus E est fini, la mesure de probabilité uniforme $(1/\operatorname{Card} E)_{x\in E}$ est invariante.*

Démonstration. — C'est immédiat ! Notons tout de même qu'on ne dit rien de l'unicité éventuelle ou non des mesures invariantes dans ce cas. □

Nous considérons toujours le cube. Les sommets ne sont pas reliés à eux-mêmes, et les coefficients diagonaux de la matrice de transition sont nuls. L'apériodicité n'est donc pas acquise. D'ailleurs on se rend compte que tous les points sont périodiques de période 2. En effet, un pas force à quitter son sommet d'origine. Pour y retourner, il est nécessaire de faire un nombre pair de pas, et on peut retourner à son origine par tout nombre pair de pas. Nous n'avons donc pas de réelle convergence vers l'équilibre.

Attaquons l'activité proprement dite.

```
// 2. Marche al\'eatoire sur le cube
// cardinal de l'espace d'\'etats k = 8;

P = [0,1,1,1,0,0,0,0; 1,0,0,0,1,1,0,0; 1,0,0,0,1,0,1,0; 1,0,0,0,0,1,1,0;
    0,1,1,0,0,0,0,1; 0,1,0,1,0,0,0,1; 0,0,1,1,0,0,0,1; 0,0,0,0,1,1,1,0]/3;

if ismarkov(P) & P == P' then
    mprintf("P est une matrice markovienne symetrique.\n");
else
    mprintf("P n''est pas une matrice markovienne symetrique.\n");
end
```

Nous avons doublé la vérification de la markoviennité de P par celle de sa symétrie. C'est tellement facile à faire que c'en est déconcertant.

Nous fixons pour toute la suite la taille des échantillons à $N = 1000$. Pour évaluer le temps de retour moyen $m_{\mathrm{r}}(i) = \mathbb{E}^i[T^i_{\mathrm{r}}] = 1/\pi\{i\} = 8$ (*voir* cours), nous nous donnons un échantillon de taille N de la variable T^i_{r}, c'est-à-dire que nous faisons N simulations de la chaîne issue de i stoppée à son temps de retour en i. Formellement, ou statistiquement, cela revient à

considérer N chaînes de Markov indépendantes, issues de i, de même matrice de transition P, et ainsi N temps d'arrêts $T_{\mathrm{r}}^{i,1}, \ldots, T_{\mathrm{r}}^{i,N}$ indépendants et d'estimer $m_{\mathrm{r}}(i)$ à l'aide de

$$\bar{T}_{\mathrm{r}}^{i} = \frac{T_{\mathrm{r}}^{i,1} + \cdots + T_{\mathrm{r}}^{i,N}}{N}$$

qui est un estimateur fortement consistant et sans biais de $m_{\mathrm{r}}(i)$. La vitesse de convergence est controlée par la variance $\mathrm{Var}(T_{\mathrm{r}}^{i})$ que nous ne connaissons pas (pour l'instant, *voir* la fin de cette section). Celle-ci est peut-être grande, la convergence lente, ... et le résultat de la simulation éloigné de la valeur théorique.

Voici un premier code destiné à seulement estimer $m_{\mathrm{r}}(i)$.

```
TDRmoyen = 0;
for n = 1:N;
    x = markovstep(1, P); t = 1;
    while x <> 1; x = markovstep(x, P); t = t+1; end
    TDRmoyen = TDRmoyen+t;
end
TDRmoyen = TDRmoyen/N;
mprintf("temps de retour moyen en i=1 : %f\n", TDRmoyen);
```

où on notera que le premier pas doit être effectué avant de commencer la boucle. Nous aurions pu nous passer de la variable auxiliaire `t` en utilisant l'associativité de l'addition pour obtenir

```
TDRmoyen = 0;
for n = 1:N;
    x = markovstep(1, P); t = 1;
    while x <> 1; x = markovstep(x, P); TDRmoyen = TDRmoyen+1; end
end
TDRmoyen = TDRmoyen/N;
mprintf("temps de retour moyen en i=1 : %f\n", TDRmoyen);
```

mais le code est alors moins clair à notre avis. De plus, la présence de la variable auxiliaire `t` permet d'obtenir facilement une estimation de la variance $\mathrm{Var}(T_{\mathrm{r}}^{i})$ et donc de l'écart-type comme suit :

```
N=1000;

// temps de retour moyen en $i=1$
TDRmoyen = 0; TDRvar = 0;
for n = 1:N;
    x = markovstep(1, P); t = 1;
    while x <> 1; x = markovstep(x, P); t = t+1; end
    TDRmoyen = TDRmoyen+t; TDRvar = TDRvar+t^2;
end
TDRmoyen = TDRmoyen/N; TDRvar = TDRvar/N-TDRmoyen^2;
mprintf("temps de retour moyen en i=1 : %f\n", TDRmoyen);
mprintf("ecart-type observe : %f\n", sqrt(TDRvar));
```

On observe généralement des temps moyens proches de la valeur théorique. Cependant, les écarts-types observés sont de l'ordre de 8, on a donc pour l'estimateur un écart-type de l'ordre de $8/\sqrt{N} \approx 1/4$ et un intervalle de confiance à 95 % d'amplitude proche de $2 \times 1{,}96 \times 8/\sqrt{N} \approx 1$, ce qui est plutôt grossier.

⚄ Le nombre moyen de visites de j avant de retourner en i est

$$\gamma^i\{j\} = \mathbb{E}\left[\sum_{n=0}^{T_{\rm r}^i-1} \mathbb{1}_{\{X_n=j\}}\right].$$

Pour i fixé, ceci definit une mesure positive qui est invariante et telle que $\gamma^i\{i\} = 1$. Par unicité de la mesure de probabilité invariante, γ^i est proportionnel à π. En particulier $\gamma^i\{o\} = \gamma^i\{i\}/\pi\{i\} \times \pi\{o\} = \gamma^i\{i\} = 1$. Nous le vérifions par simulation.

```
// Nombre moyen de visites de $o = 8$ avant de retourner en $i=1$
Vmoyen = 0; Vvar = 0;
for n = 1:N;
    x = markovstep(1, P); v = 0; // 1 <> 8, sinon v = 1
    while x <> 1;
        if x == 8 then v = v+1; end
        x = markovstep(x, P);
    end
    Vmoyen = Vmoyen+v; Vvar = Vvar+v^2;
end
Vmoyen = Vmoyen/N; Vvar = Vvar/N-Vmoyen^2;
mprintf("nombre moyen de visites de o = 8 avant de retourner en i=1 : %f\n", Vmoyen);
mprintf("ecart-type observe : %f\n", sqrt(Vvar));
```

⚄ On montre que la moyenne du temps d'atteinte T^o, $o = 8$, en étant issu de $i = 1$ est $\mathbb{E}^i[T^o] = 10$. Nous le vérifions numériquement.

```
// temps moyen avant la premi\‘ere visite de $o = 8$
TDAmoyen = 0; TDAvar = 0;
for n = 1:N;
    x = 1; t = 0;
    while x <> 8; x = markovstep(x, P); t = t+1; end
    TDAmoyen = TDAmoyen+t; TDAvar = TDAvar+t^2;
end
TDAmoyen = TDAmoyen/N; TDAvar = TDAvar/N-TDAmoyen^2;
mprintf("temps d’’atteinte moyen de o = 8 : %f\n", TDAmoyen);
mprintf("ecart-type observe : %f\n", sqrt(TDAvar));
```

Pour arriver à ce résultat théorique, en appliquant la propriété de Markov simple, nous avons formé le système linéaire

$$\begin{cases} \mathbb{E}^x\big[T^A\big] = 1 + \displaystyle\sum_{y\in E} P(x,y) \times \mathbb{E}^y\big[T^A\big] = 1 + \sum_{y\notin A} P(x,y) \times \mathbb{E}^y\big[T^A\big] & x \notin A \\ \mathbb{E}^x\big[T^A\big] = 0 \qquad x \in A \end{cases}$$

où $T^A = \inf\{n \geqslant 0 : X_n \in A\}$ est le temps d'atteinte de $A \subset E$ (ici $A = \{8\}$). Ce système s'écrit de manière matricielle

$$\big(\mathrm{Id}_E - \mathbb{1}_{A^c\times A^c} \times P\big) \times m_{\rm a} = \mathbb{1}_{A^c} \qquad \text{ou encore} \qquad m_{\rm a} = \big(\mathrm{Id}_E - \mathbb{1}_{A^c\times A^c} \times P\big)^{-1} \mathbb{1}_{A^c}$$

avec $m_{\rm a}(x) = \mathbb{E}^x\big[T^A\big]$, et est résolu par simple inversion de matrice :

```
Ac = ones(8,1); Ac(8) = 0;// vecteur compl\’ementaire de l’ensemble absorbant
mTDA = (eye(8,8)-diag(Ac)*P)\Ac;// noter ‘‘eye’’ et ‘‘diag’’ et l’inversion
```

On peut aussi déterminer les variances. Toujours par la propriété de Markov simple, si $x \notin A$,

$$\begin{aligned}
\mathbb{E}^x\big[(T^A)^2\big] &= \sum_{y\in E} P(x,y) \times \mathbb{E}^y\big[(1+T^A)^2\big] = \sum_{y\in E} P(x,y) \times \big(1 + 2\,\mathbb{E}^y\big[T^A\big] + \mathbb{E}^y\big[(T^A)^2\big]\big) \\
&= \sum_{y\in E} P(x,y) + 2\sum_{y\in E} P(x,y) \times \mathbb{E}^y\big[T^A\big] + \sum_{y\in E} P(x,y) \times \mathbb{E}^y\big[(T^A)^2\big] \\
&= 1 + 2\big(\mathbb{E}^x\big[T^A\big] - 1\big) + \sum_{y\in E} P(x,y) \times \mathbb{E}^y\big[(T^A)^2\big] \\
&= 2\,\mathbb{E}^x\big[T^A\big] - 1 + \sum_{y\in E} P(x,y) \times \mathbb{E}^y\big[(T^A)^2\big] \\
&= 2\,\mathbb{E}^x\big[T^A\big] - 1 + \sum_{y\notin A} P(x,y) \times \mathbb{E}^y\big[(T^A)^2\big]
\end{aligned}$$

et, bien sûr, $\mathbb{E}^x\big[(T^A)^2\big] = 0$ si $x \in A$. En posant $k(x) = \mathbb{E}^x[(T^A)^2]$, on obtient le système linéaire

$$\big(\mathrm{Id}_E - \mathbb{1}_{A^c\times A^c} \times P\big) \times k = 2m_{\mathrm{a}} - \mathbb{1}_{A^c}, \qquad \text{ou encore} \qquad k = \big(\mathrm{Id}_E - \mathbb{1}_{A^c\times A^c} \times P\big)^{-1}(2m_{\mathrm{a}} - \mathbb{1}_{A^c}),$$

qui est résolu une fois de plus par simple inversion de matrice :

```
k = (eye(8,8)-diag(Ac)*P)\(2*mTDA-Ac);// c'est le ''mTDA'' pr\'ec\'edent
```

Les variances sont alors données par le vecteur `k-mTDA.*mTDA`, et les écarts-types par `sqrt(k-mTDA.*mTDA)`.

Par simulation, nous procédons à l'estimation du vecteur m_{a} précédent, ainsi que des écarts-types.

```
mprintf("temps d''atteinte moyen h(i,o) de o partant de i\n");
// temps d'atteinte moyen de $o = 8$ partant de $i=1$
TDAmoyen = zeros(8,1); ; TDAcstd = zeros(8,1);
for i = 1:8;
    for n = 1:N;
        x = i; t = 0;
        while x <> 8; x = markovstep(x, P); t = t+1; end
        TDAmoyen(i) = TDAmoyen(i)+t; TDAcstd(i) = TDAcstd(i)+t^2;
    end
    TDAmoyen(i) = TDAmoyen(i)/N;
    TDAcstd(i) = sqrt((TDAcstd(i)-N*TDAmoyen(i)^2)/(N-1));
    mprintf("Vraies valeurs : m_a(%d) = %.2f, variance = %.2f, ecart-type = %.2f\n",..
        i, mTDA(i), k(i)-mTDA(i)^2, sqrt(k(i)-mTDA(i)^2));
    mprintf("Estimations :    m_a(%d) ~ %.2f, variance ~ %.2f, ecart-type ~ %.2f\n",..
        i, TDAmoyen(i), TDAcstd(i)^2, TDAcstd(i));
end
```

Ce qui aura pu donner :

```
Vraies valeurs : m_a(1) = 10.00, variance = 63.00, ecart-type = 7.94
Estimations :    m_a(1) ~ 9.90, variance ~ 72.99, ecart-type ~ 8.54
Vraies valeurs : m_a(2) = 9.00, variance = 63.00, ecart-type = 7.94
Estimations :    m_a(2) ~ 8.80, variance ~ 56.36, ecart-type ~ 7.51
Vraies valeurs : m_a(3) = 9.00, variance = 63.00, ecart-type = 7.94
Estimations :    m_a(3) ~ 8.99, variance ~ 63.08, ecart-type ~ 7.94
Vraies valeurs : m_a(4) = 9.00, variance = 63.00, ecart-type = 7.94
```

```
Estimations :    m_a(4) ~ 9.06, variance ~ 72.01, ecart-type ~ 8.49
Vraies valeurs : m_a(5) = 7.00, variance = 60.00, ecart-type = 7.75
Estimations :    m_a(5) ~ 7.14, variance ~ 58.52, ecart-type ~ 7.65
Vraies valeurs : m_a(6) = 7.00, variance = 60.00, ecart-type = 7.75
Estimations :    m_a(6) ~ 7.47, variance ~ 62.24, ecart-type ~ 7.89
Vraies valeurs : m_a(7) = 7.00, variance = 60.00, ecart-type = 7.75
Estimations :    m_a(7) ~ 6.98, variance ~ 58.95, ecart-type ~ 7.68
Vraies valeurs : m_a(8) = 0.00, variance = 0.00, ecart-type = 0.00
Estimations :    m_a(8) ~ 0.00, variance ~ 0.00, ecart-type ~ 0.00
```

On notera que pour $N = 1000$, les intervalles de confiance à 95 % des moyennes ont une amplitude de l'ordre de $2 \times 1{,}96 \times 8/\sqrt{N} \approx 1$, ce qui est assez important.

Remarque. — Nous n'en demandions pas tant en travaux pratiques. Cependant, comme nous avions affirmé ne rien connaître aux variances — ce qui était vrai —, il paraissait important d'y jeter un coup d'œil.

D'ailleurs, nous pouvons revenir aux moments du temps de retour T_{r}^i. Par application de la propriété de Markov simple au bout d'un pas, nous avons

$$\mathbb{E}^i[T_{\mathrm{r}}^i] = \sum_{j\in E} P(i,j)\,\mathbb{E}^j[T^i + 1] = 1 + \sum_{j\in E} P(i,j)\,\mathbb{E}^j[T^i]$$

où on note être passé d'un temps de retour à un temps d'atteinte. Pour $i = 8$, nous connaissons les $\mathbb{E}^j[T^8]$ (*voir* ci-dessus). On obtient alors que $m_{\mathrm{r}}(8) = \mathbb{E}^8[T_{\mathrm{r}}^8] = 1 + \frac{1}{3}(\mathbb{E}^5[T^8] + \mathbb{E}^6[T^8] + \mathbb{E}^7[T^8]) = 1 + \frac{1}{3}(7+7+7) = 8 = m_{\mathrm{r}}(i)$ comme vu précédemment. Faisons de même pour calculer la variance. On a

$$\begin{aligned}
\mathbb{E}^i\big[(T_{\mathrm{r}}^i)^2\big] &= \sum_{j\in E} P(i,j)\,\mathbb{E}^j\big[(T^i+1)^2\big] = 1 + 2\sum_{j\in E} P(i,j)\,\mathbb{E}^j[T^i] + \sum_{j\in E} P(i,j)\,\mathbb{E}^j\big[(T^i)^2\big] \\
&= 2\sum_{j\in E} P(i,j)\,\mathbb{E}^j[T^i+1] - 1 + \sum_{j\in E} P(i,j)\,\mathbb{E}^j\big[(T^i)^2\big] \\
&= 2\,\mathbb{E}^i[T_{\mathrm{r}}^i] - 1 + \sum_{j\in E} P(i,j)\,\mathbb{E}^j\big[(T^i)^2\big].
\end{aligned}$$

En se servant des résultats numériques, ceci vaut $\mathbb{E}^8\big[(T_{\mathrm{r}}^8)^2\big] = 16-1+\frac{1}{3}(109+109+109) = 124$, d'où $\mathrm{Var}^8(T_{\mathrm{r}}^8) = \mathrm{Var}^i(T_{\mathrm{r}}^i) = 124 - 64 = 60$ et un écart-type voisin de 7,75 comme les simulations l'avaient déjà suggéré.

3. **Le contrôle des naissances**

Rien à dire.

4. **L'urne d'Ehrenfest**

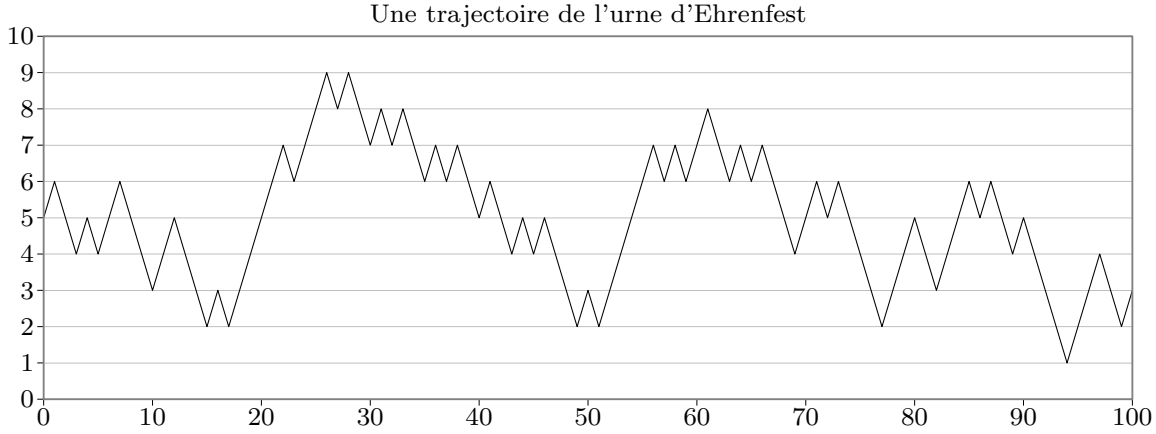
Beaucoup de choses ont été vues sur l'urne d'Ehrenfest en cours et en travaux dirigés. Il ne restait plus qu'à l'aborder de manière numérique pour faire quelques graphiques notamment.

```
// 3. L'urne d'Ehrenfest

clear; mode(0); usecanvas(%f);

function y = ehrenstep(x, n);
    if rand() < x/n then y = x-1; else y = x+1; end
endfunction
```

On note qu'il serait inutilement compliqué et excessivement rigide de se donner n le nombre de boules, de définir la matrice de transition P de dimensions $n \times n$ associée, et de simuler la chaîne via la commande `markovstep(`$\langle x \rangle$`,`$\langle P \rangle$`)`. La petite instruction `ehrenstep(`$\langle x \rangle$`,`$\langle P \rangle$`)` définie ci-dessus convient très bien.

Nous fixons ci-dessous quelques paramètres : n le nombre total de boules, x_0 une valeur initiale entière et T l'horizon temporel. La simulation d'une trajectoire et son tracé sont immédiats.

```
n = 10; xzero = floor(n/2); T = 100;

x = xzero; for i = 1:T; x = [x; ehrenstep(x(i),n)]; end
scf(1); clf(1); plot([0:T]', x);
xtitle("Une trajectoire de l''urne d''Ehrenfest");
```



Pour la suite, nous nous donnons une taille d'échantillons $N = 1000$. Les lois observées, ou empiriques, seront donc toutes basées sur N observations.

Nous considérerons 8 lois. Tout d'abord les lois théoriques : la loi invariante $\mathcal{B}(n, 1/2)$, sa partie paire (loi conditionnelle à un résultat pair), sa partie impaire (loi conditionnelle à un résultat impair), la moyenne de ces deux dernières lois (qui est différente en général de la loi binomiale considérée). Puis les lois empiriques correspondantes : les lois trajectorielles convergent presque sûrement vers la loi invariante d'après le théorème ergodique, la convergence vers l'équilibre à lieu pour les sous-suites respectivement d'indices pairs et impairs, d'où la convergence des lois moyennes.

Nous stockons toutes les lois sous la forme d'une matrice (une ligne correspond à une loi) en gardant dans un vecteur la description de celles-ci. On prendra garde au fait que le support de ces lois est $\{0, 1, \ldots, n\}$ alors que les indices correspondants pour les vecteurs lignes sont $\{1, 2, \ldots, n+1\}$.

```
N = 1000; // taille des \'echantillons

nomloi = ["loi B(n,1/2)";
    "partie paire";
    "partie impaire";
    "loi moyenne";
    "loi trajectorielle";// voir le th\'eor\`eme ergodique
    "loi empirique au temps T";// voir la convergence vers l'\'equilibre
    "loi empirique au temps T+1";// voir la convergence vers l'\'equilibre
    "loi empirique moyenne"];// voir la convergence vers l'\'equilibre

loi = zeros(8, n+1);

// loi B(n,1/2), partie paire, partie impaire, loi moyenne
```

```
loi(1,1)=1; for i = 1:n; loi(1,i+1) = loi(1,i)*(n-i+1)/i; end
for i = 0:n;
    if i/2 == floor(i/2) then loi(2,i+1) = loi(1,i+1);
    else loi(3,i+1) = loi(1,i+1); end
end
loi(1,:) = loi(1,:)/2^n;
loi(2,:) = loi(2,:)/sum(loi(2,:));
loi(3,:) = loi(3,:)/sum(loi(3,:));
loi(4,:) = (loi(2,:)+loi(3,:))/2;
```

Voici pour les lois théoriques. Passons aux simulations.

```
// lois aux dates T et T+1 et loi moyenne

for j = 1:N;
    x = xzero;
    for i = 1:T; x = ehrenstep(x,n); end
    loi(6,x+1) = loi(6,x+1)+1;
    x = ehrenstep(x, n);
    loi(7, x+1) = loi(7,x+1)+1;
end
loi(6,:) = loi(6,:)/N;
loi(7,:) = loi(7,:)/N;
loi(8,:) = (loi(6,:)+loi(7,:))/2; // attention N observations;
```

Notons que pour la loi trajectorielle l'horizon temporel n'est plus T ou $T+1$ qui peut s'avérer trop petit pour nos ambitions, mais la taille N d'échantillons.

```
// loi trajectorielle

x = xzero; loi(5,x+1) = 1;
for i = 2:N; x = ehrenstep(x,n); loi(5,x+1) = loi(5,x+1)+1; end
loi(5,:) = loi(5,:)/N; // normalisation
```

La parité du support de la loi de X_T dépend de la parité de T et de celle de x_0. Il faut peut-être échanger quelques vecteurs en fonction de celles-ci.

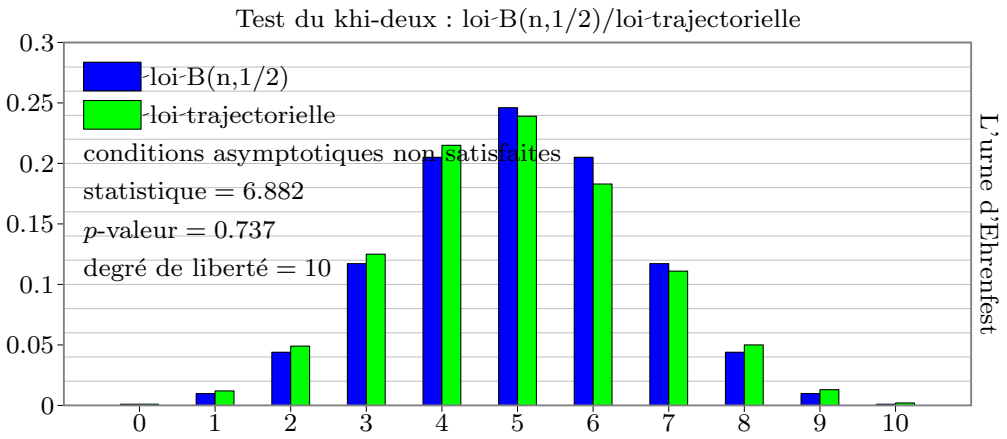
```
// ajustement de parit\'e

if (xzero+T)/2 <> floor((xzero+T)/2) then
    tmp = loi(6,:); loi(6,:) = loi(7,:); loi(7,:) = tmp;
    tmp = nomloi(6); nomloi(6) = nomloi(7); nomloi(7) = tmp;
end
```

Après s'être beaucoup servi d'une procédure « clef en main » (`dgoftest()`), revenons aux fondamentaux pour tracer des diagrammes en bâtons et faire des tests du χ^2 d'adéquation (nous ferons quelques commentaires sur la légitimité de certains d'entre eux). On notera que tout a été préparé pour que les 4 comparaisons puissent se faire à l'aide d'une simple boucle...

```
// diagrammes en b\^atons et tests du khi-deux

chisquare = zeros(4,1);
dl = -ones(4,1);
pvaleur = zeros(4,1);
```

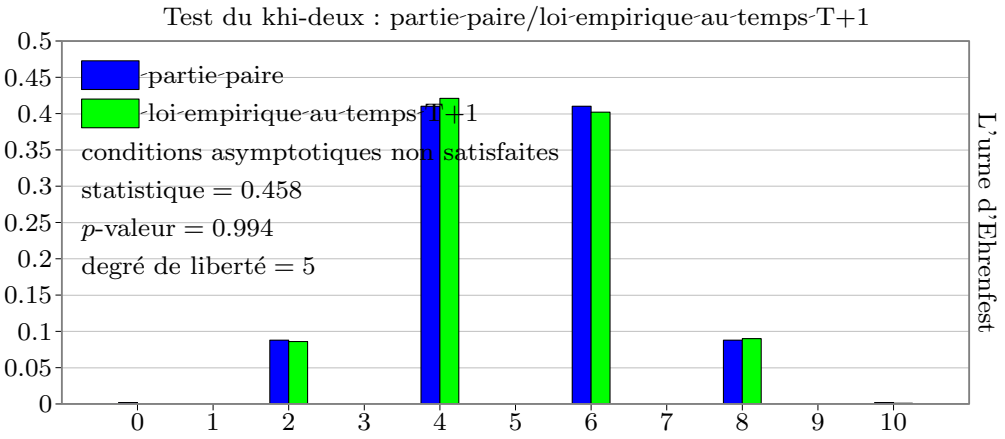
```
for i = 1:4;
    scf(i+1); clf(i+1);
    bar([0:n]',[loi(i,:)',loi(i+4,:)']);// attention aux transpositions
    xtitle("L''urne d''Ehrenfest");
    legend([nomloi(i); nomloi(i+4)]);
    flag = %t;
    for j = 1:n+1;
        if loi(i,j) > 0 then
            dl(i) = dl(i)+1;
            chisquare(i) = chisquare(i)+(loi(i+4,j)-loi(i,j))^2/loi(i,j)*N;
            flag = flag & (N*loi(i+4,j) >= 5);
        end
    end
    pvaleur(i) = 1-cdfchi("PQ", chisquare(i), dl(i));
    mprintf("Test du khi-deux : %s/%s\n", nomloi(i), nomloi(i+4));
    mprintf("statistique = %.3f\n", chisquare(i));
    if flag then mprintf("conditions asymptotiques satisfaites\n");
    else mprintf("conditions asymptotiques non satisfaites\n"); end
    mprintf("p-valeur = %.3f (degre de liberte = %d)\n", pvaleur(i), dl(i));
end
```

Les graphiques suivants sont comparables à ceux qu'on obtient avec SCILAB. Le commentaire est général et ne correspond pas forcément à l'illustration.

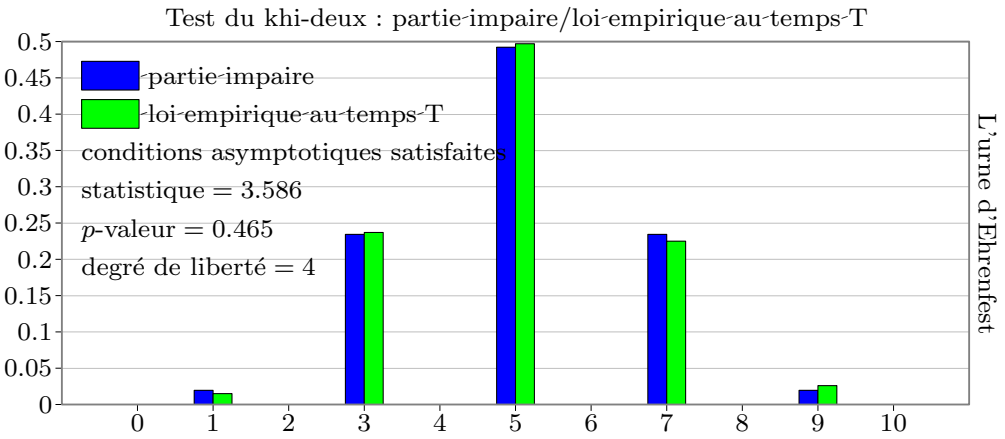


Toutes les hypothèses du théorème ergodique sont satisfaites : l'irréductibilité et plus encore la récurrence positive. Ainsi, presque sûrement les lois trajectorielles convergent vers la loi invariante. Ici, le résultat est parfois médiocre. Le temps limite $N = 1000$ est peut-être trop petit. De plus, on doit noter que les observations ne sont pas indépendantes puisque nous suivons une trajectoire. Le test du χ^2 ne devrait pas être employé, du moins, sa p-valeur

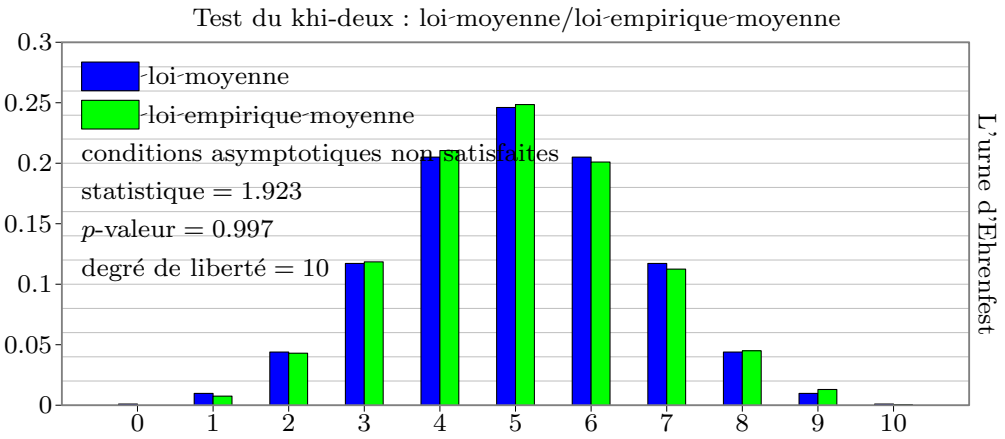
devrait être ignorée.



Au temps $T = 100$, l'adéquation de la loi empirique avec la partie de parité correspondante de la mesure invariante est généralement bien vérifié. Noter que le temps T est vu comme grand, c'est-à-dire qu'on s'intéresse à un comportement asymptotique en temps.



Nous avons les mêmes remarques ici.



Pour les moyennes sur deux instants consécutifs, l'adéquation ne peut être que meilleure encore. Le test du χ^2 est aussi fantaisiste que pour l'étude sur une trajectoire : nous avons ici $2n$ observations liées deux à deux. Nous aurions dû procéder de manière indépendante pour les échantillons aux temps T et au temps $T + 1$. Alors, on aurait dû multiplier la statistique observée par 2 et recalculer la p-valeur associée...

Table des matières

1. L'aiguille de Buffon 1
2. Méthode de Monte Carlo pour le calcul approché d'intégrales 3
3. Méthode de Monte Carlo pour le calcul approché d'un volume 4
Références 5
1. L'aiguille de Buffon 10
2. Méthode de Monte Carlo pour le calcul approché d'intégrales 11
3. Méthode de Monte Carlo pour le calcul approché d'un volume 14
1. Généralités 15
2. Marche aléatoire sur le cube 17
3. Le contrôle des naissances 18
4. L'urne d'Ehrenfest 19
1. Généralités 20
2. Marche aléatoire sur le cube 22
3. Le contrôle des naissances 27
4. L'urne d'Ehrenfest 27