# Modali 2005

Jean-Philippe Chancelier and Michel de Lara

October 10, 2017

## Contents



## 1 Random walk on $\mathbb{Z}$

Let $(Y(t))_{t\in\mathbb{N}}$ be a sequence of i.i.d. random variables having values in $\{-1,+1\}$ as follows:

$$\mathbb{P}(Y(t)=1)=p \quad \text{and} \quad \mathbb{P}(Y(t)=-1)=1-p\,. \tag{1}$$

We define a stochastic process $(X(t))_{t\in\mathbb{N}}$ on $\mathbb{Z}$ called *random walk* by

$$X(0)=0 \quad \text{and} \quad X(t+1)=X(t)+Y(t)\,. \tag{2}$$

**Question 1** *Simulate one single trajectory for $p=0.5$.*

```
T= 100;
time=1:(T+1);
p=0.5;
trajectory=zeros(time) ;
// vector will contain the trajectory X(0),...,X(T)
for t=1:T
trajectory(t+1)=trajectory(t) + ( 2*grand(1,1,'bin',1,p)-1 ) ;
```

```
end ;
xset("window",1); xbasc(1); plot2d2(time,trajectory);
xtitle('Random walk (p='+string(p)+')', 'time (t)','X(t)')
```

**Question 2** *Draw four trajectories for $p = 0.5$ on the same picture.*

```
S=4;
trajectory=zeros(S,sum(prod(ones(time))));
// vector will contain the S trajectories X(0),...,X(T)
for s=1:S,
for t=1:T
trajectory(s,t+1)=trajectory(s,t) + ( 2*grand(1,1,'bin',1,p)-1 ) ;
end ;
end
xset("window",2); xbasc(2); plot2d2(time',trajectory(1:S,:)' );
xtitle('Random walk (p='+string(p)+')', 'time (t)','X(t)')
```

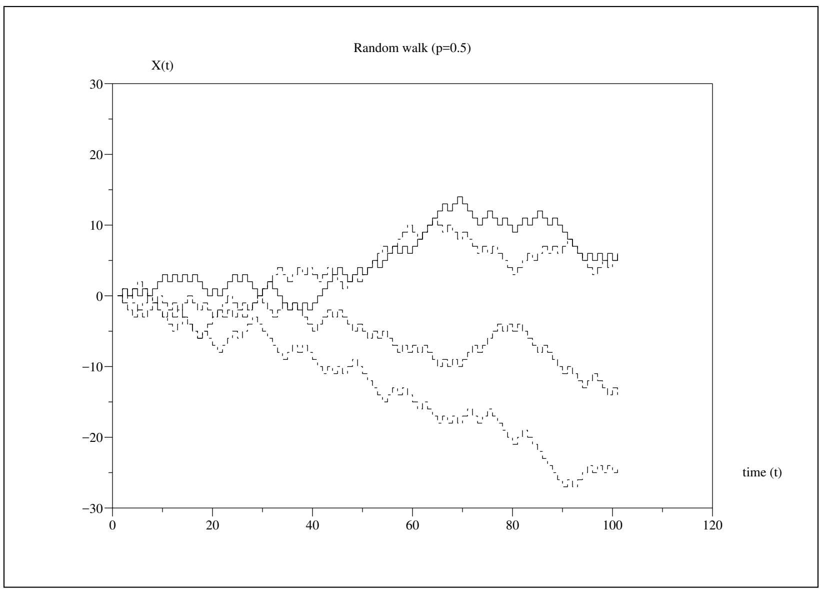


Figure 1: *Four trajectories of the symmetric random walk on $\mathbb{Z}$*

**Question 3** *Take $p \neq 0.5$ and redo simulations.*

## 2 Random walk on $\mathbb{Z}^2$

Let $(Y(t))_{t\in\mathbb{N}}$ be a sequence of i.i.d. random variables having values in $\{-1,+1\}^2$ as follows:

$$\mathbb{P}(Y(t)=(\epsilon,\epsilon'))=p_{\epsilon,\epsilon'} \quad \text{with} \quad \sum_{(\epsilon,\epsilon')\in\{-1,+1\}^2} p_{\epsilon,\epsilon'}=1\,. \tag{3}$$

We define a stochastic process $(X(t))_{t\in\mathbb{N}}$ on $\mathbb{Z}^2$ called 2-dimensional *random walk* by

$$X(0)=0 \quad \text{and} \quad X(t+1)=X(t)+Y(t)\,. \tag{4}$$

**Question 4** *Simulate trajectories for* $p_{\epsilon,\epsilon'}=0.25$.

```
T= 100;
time=1:(T+1);
p=0.5;
trajectory=zeros(2,T+1) ;
// vector will contain the trajectory X(0),...,X(T)
for t=1:T
trajectory(:,t+1)=trajectory(:,t) + ( 2*grand(2,1,'bin',1,p)-1 ) ;
end ;
xset("window",1);xbasc(1); plot2d(trajectory(1,:),trajectory(2,:),...
style=3,...
rect=[-maxi(abs(trajectory(1,:))),-maxi(abs(trajectory(2,:))),...
maxi(abs(trajectory(1,:))),maxi(abs(trajectory(2,:)))] );
xtitle('Symmetric random walk', 'time (t)','X(t)')
```

## 3 Transmitting 0 and 1

$$\begin{cases} \mathbb{P}(X(t+1)=1\,|\,X(t)=1) &= \mathbb{P}(X(t+1)=0\,|\,X(t)=0) &= p \\ \mathbb{P}(X(t+1)=1\,|\,X(t)=0) &= \mathbb{P}(X(t+1)=0\,|\,X(t)=0) &= 1-p \end{cases} \tag{5}$$

may be simulated by

$$X(t+1)=X(t)Y(t)+(1-X(t))(1-Y(t))=Y(t)\,(2X(t)-1)+1-X(t) \tag{6}$$

where $(Y(t))_{t\in\mathbb{N}}$ is a sequence of i.i.d. random variables with Bernoulli law $\mathcal{B}(p;1)$.

```
T= 100;
time=1:(T+1);
p=0.9;
trajectory=zeros(time) ;
// vector will contain the trajectory X(0),...,X(T)
```

```
for t=1:T
trajectory(t+1)= (2*trajectory(t)-1)*grand(1,1,'bin',1,p) + ...
1- trajectory(t)
end ;
xset("window",1); xbasc(1); plot2d2(time,trajectory, rect=[0,-0.1,T,1.1]);
xtitle('0 or 1 (p='+string(p)+')', 'time (t)','X(t)')
```

# 4 Testing the irreductibility of a Markov chain

On utilise Scilab pour visualiser le graphe associé à une chaîne de Markov :

```
 M  = [    0.1    0.7    0.     0.1    0.     0.1
           0.     0.3    0.7    0.     0.     0.
           0.     0.4    0.6    0.     0.     0.
           0.     0.     0.     0.7    0.     0.3
           0.     0.8    0.     0.     0.1    0.1
           0.     0.     0.     0.8    0.     0.2 ];
 Mb=M<>0 ; //  Matrice Booléene associée
 g=mat_2_graph(sparse(bool2s(Mb)),1,'node-node'); //  Graphe  associé
 n = size(M,'r');
 theta = (2*%pi/n)*(1:n);
 g('node_x')=sin(theta)*100;
 g('node_y')=cos(theta)*100;
 show_graph(g);
```

On peut trouver les classes de la chaîne de Markov en cherchant les composantes (fortement) connexes du graphe associé.

```
[nc,ncomp]=strong_connex(g); // Calcul des composantes connexes
```

**Question 5** 1. *Dans les composantes connexes recherchez celles qui correspondent à des classes finales.*

2. *Construire la permutation des états qui permet d'écrire* `M` *sous la forme canonique où on voit tout de suite les classes finales et la classe transitoire :*

```
Mc=M(perm,perm)  // Forme canonique de M.
 Mc  =

!   0.2    0.8    0.     0.     0.     0.  !
!   0.3    0.7    0.     0.     0.     0.  !
!   0.     0.     0.3    0.7    0.     0.  !
!   0.     0.     0.4    0.6    0.     0.  !
!   0.1    0.     0.8    0.     0.1    0.  !
!   0.1    0.1    0.7    0.     0.     0.1 !
```

# 5 Une option européenne discrète

La marche aléatoire de Cox Ross est définie de la façon suivante : soient $a$ et $b$ deux nombres réels et soit $(U_n)_{n\geq 0}$ une suite de variables aléatoires indépendantes de loi commune :

$$U_n = \begin{cases} 1+a & \text{avec probabilité} \quad p \\ 1+b & \text{avec probabilité} \quad 1-p \end{cases}$$

avec $0 < p < 1$. On défini alors une suite de variables aléatoires $(S_n)_{n\in\mathbb{N}}$ par :

$$S_1 = s_1 \quad \text{et} \quad S_{n+1} = S_n U_{n+1}.$$

On a vu en cours que la suite $S_n$ est une chaîne de Markov : Supposons que l'on s'intéresse au temps $n \in [0, N]$, comme $S_n = s_1 \prod_{i=1}^n U_i$, l'espace d'état de la chaîne $(S_n)_{n\in\mathbb{N}}$ est alors fini, $\mathcal{E} = \{\gamma_{i,j} \equiv s_1(1+a)^i(1+b)^j, \quad 0 \leq i+j < N\}$. On vérifie que $S_n$ est une chaîne de Markov homogène de matrice de transition:

$$M_{\gamma_{i,j},\gamma_{k,l}} = \begin{cases} p & \text{si} \quad (k,l) = (i+1,j) \\ 1-p & \text{si} \quad (k,l) = (i,j+1) \\ 0 & \text{sinon} \end{cases} \tag{7}$$

La suite $S_n$ précédemment définie peut servir à modéliser le cours d'un actif. On cherche à calculer pour $s_1$ fixé la valeur :

$$v_1(s_1) = \mathbb{E}\left[\frac{1}{(1+r)^N} f(S_N) | S_1 = s_1\right]. \tag{8}$$

On a vu en cours que cette quantité peut se calculer de façon récursive en calculant pour $n$ allant de $N$ à 0 les quantités :

$$v_n(x) = \mathbb{E}\left[\frac{1}{(1+r)^{N-n}} f(S_N) | S_n = x\right]. \tag{9}$$

On se propose dans cet exercice de calculer $v_n(x)$ récursivement pour obtenir $v_1(s_1)$. On utilisera les données suivantes :

```
a= 0.05;
b= -0.05;
s0=1 ;
r=0.01;
pa = (r-a)/(b-a);
pb = (b-r)/(b-a);
```

Les états possibles de la chaîne sont indicés par un couple $(i,j)$ avec $i \in [0,N]$ et $j \in [0,N]$ et on veut dans Scilab utiliser une matrice $V$ pour stocker la fonction $v(x,n)$. Il faut donc pouvoir coder les états avec un unique indice et on retiendra $i + N * j + 1$. La matrice utilisée dans Scilab sera de taille $V(N^2, N)$ (on ne cherche pas dans un premier temps à économiser de la place mais on privilégie du calcul matriciel).

**Question 6** *1. Construire une matrice* `s(N,N)` *qui contient les états possibles de la chaîne. Utiliser la fonction* `matrix` *pour transformer* `s` *en un vecteur colonne de taille* `NxN`.

2. *Initialiser une matrice* `V` *de taille* `(NxN,N)`.

3. *On suppose que $f(x) = (x-1)^+$. Initialiser la dernière colonne de* `V` *(*`V(:,N)`*) avec la fonction $f$ et en utilisant le vecteur colonne contenant les états possibles.*

4. *calculer récursivement* `V(:,n)` *en utilisant les résultats du cours. On notera que pour certains états à l'instant $n$ les états possibles à l'instant $n+1$ sont hors de l'ensemble* `[1,N]` *mais ce n'est pas grave car la valeurs de* `V(:,n+1)` *en ces points n'interviennent pas dans le calcul de $v_1(s_1)$. On pourra donc utiliser zéro comme valeurs de $V$ en ces points.*

5. *Pour les points atteignables à partir de $s_1$ tracer les courbes $V(:,n)$.*

# 6 Le problème de la secrétaire

On rappelle ici la formulation vue en cours. On dispose de $N$ objets de valeurs $v_k$ ( $v_1 < v_2 < \ldots < v_N$ ) inconnues que l'on tire au hasard (de façon équiprobable) les uns après les autres. Quand on tire le $k$-ième objet on peut soit le choisir et s'arrêter soit continuer. Les objets que l'on va tirer successivement auront pour valeur $v_{\sigma(k)}$ où $\sigma$ est une permutation de $[1,N]$ et toutes les suites $(v_{\sigma(k)})_{k=1,N}$ sont équiprobables.

Introduisons les notations suivantes :

$$S_k = \begin{cases} 1 & \text{si} \quad W_k = k \quad \text{avec} \quad W_k = \arg\max_{j\in[1,k]} v_{\sigma(j)} \\ 0 & \text{sinon} \end{cases}$$

On a vu que la suite $S_n$ est une chaîne de Markov. L'espace d'état de $S_k$ est $\{0,1\}$ et les Matrices de transition $M^{(k)}$ vérifient : $M^{(k)}_{i,1} = 1/(k+1)$ et $M^{(k)}_{i,0} = k/(k+1)$ (les $S_k$ sont en fait indépendants).

Le problème du décideur est un problème d'arrêt optimal. Il cherche à calculer $u_1(1)$ et obtenir la stratégie optimale associée (noter que $S_1 \equiv 1$), où $u_n(x)$ est donné par la formule générale :

$$u_n(x) \equiv \sup_{\tau\ \mathcal{F}_n t.a., n\le\tau\le N} \mathbb{E}\left[g_\tau(S^\tau)|S_n = x\right].$$

avec $g_k(1) = k/N$ et $g_k(0) \equiv 0$.

On rappelle que $u_n(x)$ est solution de l'équation suivante :

$$u_n(x) = \max\left(\frac{1}{n+1}u_{n+1}(1) + \frac{n}{n+1}u_{n+1}(0), \frac{n}{N}\mathbb{I}_{x\neq 0}\right) \quad ; u_N(x) = g_N(x) = x \tag{10}$$

**Question 7** *1. Écrire un programme qui calcule la fonction $u_n(x)$ pour $n \in [1,N]$ et tracer les courbes $u_n(1)$ et $u_n(0)$ ainsi que les fonctions $g_n(1)$ et $g_n(0)$.*

2. *On suppose que les valeurs des secrétaires sont* $[1, N]$*. Utiliser* `grand` *pour en tirer une permutation aléatoire.*

3. *Calculer le long de cette trajectoire la valeur de* $u_n$ *et trouver le temps d'arrêt i.e l'indice de la secrétaire choisie.*

4. *Illustrer graphiquement les résultats précédents.*