HEC 922, 2001/2002 **TP 4 - CORRIGE**

17, 23 et 24 octobre 2001

Simulation d'expériences probabilistes

Laure Danthony
http://www.ens-lyon.fr/~ldanthon/

1 HEC

- On utilise le modèle suivant :
  – A vaut 0 si $A \in C_0$, 1 sinon.
  – B vaut 0 si $B \in C_0$, 1 sinon.
  A chaque tirage, on écrira donc le couple (A, B). Pour réaliser un tirage, on fait bien sûr appel à la fonction `1+random(3)` qui retourne un entier pris dans l'ensemble $\{1, 2, 3\}$. On assimile a à 1, b à 2, c à 3.
- On en déduit immédiatement le code suivant :

```
program simul;

procedure tirage_suivant(var A,B : integer);
var result : integer;
begin
   result:=1+random(3);
   if (result=1) then A:=1-A else
      if (result=2) then B:=1-B
end;

var X,Y,nb_tirages,i:integer;

BEGIN
   randomize;
   X:=0;Y:=0;
   writeln('nb tirages ?');
   readln(nb_tirages);
   for i:=1 to nb_tirages do
   begin
      tirage_suivant(X,Y);
      writeln('(X,Y)=(',X,',',Y,')');
   end;
   readln;
END.
```

- Quelques remarques :
  – On demande bien à l'utilisateur combien de tirages il veut réaliser.
  – La procédure `tirage_suivant` réalise l'effet d'un tirage sur le couple (A, B). Les valeurs de A et B sont éventuellement modifiées (on a utilisé le mot clef `var`) suivant la valeur de `1+random(3)` et les valeurs

précédentes de A et de B. L'opération $1-A$ réalise en fait un changement d'urne pour le jeton A (pourquoi ?).

– Il convient de distinguer l'effet du tirage sur les jetons A et B de l'impression de cet effet à l'écran. D'où le fait que l'on écrit un `writeln` qu'à l'intérieur du corps du programme (et pas dans la procédure `tirage_suivant`).

2 AGRO

A. Un premier essai de simulation

A.1 Cette probabilité P_∞ est la probabilité pour que l'on ait tiré 99 fois la valeur de X qui a été fixée une fois pour toutes au début. D'où

$$P_\infty = \Big(\frac{1}{N}\Big)^{99}.$$

A.2
- Le nombre d'appels à la fonction `random` est situé entre 2 et 100.
- Calculons $P(T=2)$: c'est la probabilité pour que l'on tire $Y \neq X$ au deuxième coup :

$$P(T=2) = \frac{N-1}{N} = 1 - \frac{1}{N}.$$

Ensuite, $P(T=3)$ est la probabilité pour que l'on obtienne $Y = X$ au deuxième coup, puis $Y \neq X$ au troisième coup. Donc

$$P(T=3) = \frac{1}{N}\Big(1-\frac{1}{N}\Big).$$

De même, on obtient pour $k \leq 99$,

$$P(T=k) = \Big(\frac{1}{N}\Big)^{k-2}\Big(1-\frac{1}{N}\Big).$$

Enfin, pour $T = 100$, on a obtenu $Y = X$ 98 fois, donc

$$P(T=100) = \Big(\frac{1}{N}\Big)^{98}.$$

- On vérifie que la somme des probabilités est 1, avec MAPLE :

```
> simplify(sum((1/N)^(k-2)*(1-1/N),k=2..99)+(1/N)^98);

                                                1
```

Ouf !

- Pour l'espérance, vérifier que votre calcul est correct avec MAPLE :

```
> simplify(sum(k*(1/N)^(k-2)*(1-1/N),k=2..99)+100*(1/N)^98);

                2           97      98        /   98
(1 + N + N + ... + N    + 2N     )  /  N
                                      /
```

C'est pas beau ! Mais si on fait :

```
> limit(",N=infinity);
```

2

Ne serait-ce pas ce que l'on veut ?

A.3 Soit $i \neq j$. La probabilité pour que $X = i$ à l'issue de la procédure est la même qu'au début, à savoir $\frac{1}{N}$. Pour la variable Y, il faut distinguer les cas de sortie de la procédure :

– Soit on est sorti lorsque $I = 100$, et alors on a une chance sur N de tomber sur un j donné : $P = P_\infty . \frac{1}{N}$;

– Soit on a réalisé des tirages du type $i \dots ij$ (nombre de i inférieur strict à 98) : $P = (1 - P_\infty)\frac{1}{N-1}$

D'où en tout

$$P(Y = j) = \Big(\frac{1}{N}\Big)^{99} . \frac{1}{N} + \Big(1 - (\frac{1}{N})^{99}\Big)\Big(\frac{1}{N-1}\Big)$$

Comme les variables X et Y sont indépendantes, on a donc :

$$P(X = i, Y = j) = \frac{1}{N}\Big[\Big(\frac{1}{N}\Big)^{99} . \frac{1}{N} + \Big(1 - (\frac{1}{N})^{99}\Big)\Big(\frac{1}{N-1}\Big)\Big]$$

A.4 Dans l'expérience de tirage sans remise, la probalité d'obtenir le couple (i, j) dans cet ordre est $\frac{1}{N}\frac{1}{N-1}$. Le programme `TRUC1` réalise donc une bonne approximation de tirage sans remise pour N grand[1].

B. Un exemple de simulation d'un tirage sans remise

B.1 Cette procédure réalise l'échange entre `Tb[Z]` (la valeur située dans la case numéro Z du tableau) et la valeur située dans une des cases précédentes i ($1 \leq i \leq Z$) de ce même tableau, le tirage de cette case i étant effectuée au hasard selon la distribution uniforme.

B.2 On vérifie facilement que le tableau Tb a pris les différentes valeurs suivantes :

$$\boxed{1\,|\,2\,|\,3\,|\,4\,|\,5} \rightarrow \boxed{1\,|\,2\,|\,3\,|\,5\,|\,4} \rightarrow \boxed{1\,|\,5\,|\,3\,|\,2\,|\,4}$$
$$\rightarrow \boxed{1\,|\,3\,|\,5\,|\,2\,|\,4} \rightarrow \boxed{3\,|\,1\,|\,5\,|\,2\,|\,4}$$

B.3 On cherche dans un premier temps qu'elle était la probabilité d'obtenir `[3,1,5,2,4]` avec la procédure précédente : est-ce qu'on aurait pu obtenir avec la même procédure, mais avec des tirages de X différents, la même permutation de $[\![1, 5]\!]$?

- Après le passage à $Z = 5$, la valeur de `Tb[5]` ne sera plus modifiée, donc il faut obtenir 4 au premier tirage aléatoire (en fait, la *position* de 4, c'est-à-dire 4) ;

[1]Regardez ce qui se passe pour N grand dans l'expression du A.3

- De même, après $Z = 4$, la valeur de `Tb[4]` ne sera plus modifiée, donc il faut obtenir $X = 2$ (la position du chiffre 2 à cet instant) au deuxième tirage ;
- Après la 3-ème étape ($Z = 3$), `Tb[3]` ne sera plus modifiée, donc il est nécessaire que X prenne la valeur de la position de 5 à cet instant ;
- Et ainsi de suite ...

On vient d'établir qu'à chaque étape (à chaque valeur de Z), on avait une seule possibilité pour choisir X entre 1 et Z. La probabilité d'obtenir cette permutation (*avec cet algorithme*) était donc : $\frac{1}{5} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{5!}$

Le cas général n'est pas plus difficile à traiter. Pour $Z = N$, on doit tirer $X = x_N$ (qui est un entier entre 1 et N d'où proba $\frac{1}{N}$). Pour $Z = N - 1$, on doit tirer la position de x_{N-1} (proba $\frac{1}{N-1}$), et ainsi de suite... Finalement, la probabilité d'obtenir des tirages de X aboutissant au tableau donné est $\frac{1}{N} \cdot \frac{1}{N-1} \cdot \frac{1}{N-2} \cdots \frac{1}{2}$, soit $\frac{1}{N!}$. Cet algorithme est donc un bon algorithme pour tirer au sort des permutations aléaoires. Il les tire de façon équiprobable.

Mais, quel est le rapport avec le titre de l'exercice ? Si on considère que l'on tire des jetons sans remise, en vidant entièrement le sac, l'algorithme précédent donne des tirages équiprobables. Si on veut seulement tirer 2 jetons parmi N sans remise, on fait tourner l'algorithme précédent, et on prend comme résultat les 2 premières cases du tableau ...