

Travaux Pratiques : séance n°6

Module de Génie Informatique en Licence 3 d'électronique

Laboratoire LTSI - UMR INSERM 642 - Université de Rennes1

Nous simulerons au cours de ce TP le jeu du LOTO de la Française des Jeux. Plus exactement, nous produirons sous MATLAB un programme permettant de tirer au hasard K boules ($K = 7$ pour le LOTO) dans une urne contenant initialement N boules ($N = 49$ pour le LOTO) portant chacune un numéro différent. Nous considérerons dans un premier temps le cas d'un tirage avec remise. Puis nous nous intéresserons au cas d'un tirage sans remise caractérisant en particulier le jeu du LOTO.

Générer une suite de $N = 49$ nombre allant de 1 jusqu'à 49 et les stocker dans un vecteur ligne nommé X .

1 Tirage avec remise

La programmation du tirage avec remise peut être réalisée très simplement. En effet il suffit de calculer K réalisations d'une variable aléatoire discrète équirépartie dans l'ensemble $\{1, 2, 3, \dots, N\}$.

Montrer qu'il est possible d'obtenir K réalisations d'une variable aléatoire discrète équirépartie dans l'ensemble $\{1, 2, 3, \dots, N\}$ à partir de K réalisations d'une variable aléatoire continue de loi uniforme sur l'intervalle $[0, 1]$. En déduire un code de programmation sous MATLAB.

2 Tirage sans remise

Nous allons résoudre ce problème de deux manières distinctes. La première, bien que présentant le grand intérêt d'être intuitive, s'avère être plus coûteuse en temps de calcul que la seconde.

En effet, une façon intuitive de traiter le problème consiste à :

étape 1 : tirer au hasard une boule parmi l'ensemble des N boules de l'urne, noter son numéro X_n et replacer la boule dans l'urne ;

étape 2 : tirer au hasard dans l'urne une autre boule, portant le numéro X_m ;

étape 3 : comparer X_n et X_m : si égalité se contenter de remettre la boule dans l'urne sans noter son numéro, sinon noter son numéro après remise dans l'urne ;

étape 4 : retourner à l'étape 1 si K numéros de boules n'ont pas encore été notés, sinon s'arrêter.

Implémenter sous Matlab la méthode précédente de manière à tirer au hasard et sans remise $K=7$ nombres distincts du vecteur X . Utiliser pour cela les fonctions "find" et "isempty" de MATLAB.

Il va de soi que les chances de retomber sur une boule déjà tirée augmentent au fur et à mesure des tirages. Ceci a pour conséquence d'accroître le temps de calcul de la méthode.

Utiliser la fonction "cputime" afin de mesurer le temps nécessaire à l'exécution du programme précédent et représenter graphiquement les variations de ce temps en fonction du paramètre K pour $N=49$ fixé.

En ce qui concerne la seconde méthode, elle consiste à :

étape 1 : tirer au hasard une boule parmi l'ensemble des N boules de l'urne en repérant sa position dans l'urne, noter son numéro, puis remettre la boule dans l'urne en échangeant sa place avec celle de la boule située tout au fond de l'urne ;

étape 2 : tirer au hasard dans l'urne une autre boule parmi les $N-1$ boules du dessus, noter son numéro et la position qu'elle avait dans l'urne, puis remettre la boule dans l'urne en échangeant sa place avec celle de la boule située juste au-dessus de la dernière boule ;

(...)

étape k : tirer au hasard dans l'urne une autre boule parmi les $N-k+1$ boules du dessus, noter son numéro et la position qu'elle avait dans l'urne, puis remettre la boule dans l'urne en échangeant sa place avec celle de la boule située juste au-dessus des $k-1$ dernières boules ;

(...)

étape K : tirer au hasard dans l'urne une dernière boule parmi les $N-K+1$ boules du dessus et noter son numéro.

Implémenter sous Matlab la méthode précédente de manière à tirer au hasard et sans remise $K=7$ nombres distincts du vecteur X , puis comparer le temps d'exécution de ce programme avec le programme précédent.