

03/06/2015

Morgan LIENARDY

Jihade TIKA

IMR2

**Simulations numériques**

Probabilités et Statistiques

# Exercice 1 : Simulation des lois de probabilité

## Partie 1 :

### Générer N nombres aléatoires de loi uniforme sur [0,1]

Le but est de générer une séquence de nombres de façon pseudo-aléatoire selon une distribution uniforme.

Plus de détails sont dans les commentaires du code.

### Représenter la séquence par un histogramme pour chaque N

Plus N augmente, plus l’histogramme tend à s’équilibrer au niveau des valeurs. On observe de moins en moins de discontinuité. Car en augmentant N, on a moins de chance d’avoir des valeurs loin de la réalité. On peut supposer qu’en choisissant N infini, les valeurs de l’histogramme seraient égales entre elles.

### Utiliser le test du Chi2 pour caractériser ces séquences générées

Posons l’hypothèse H0 : « la séquence donnée en paramètres suit la loi Uniforme sur [0,1] ».

On a la probabilité p = 1/n selon la loi uniforme discrète.

Pour simplifier notre test, nous répartissons les valeurs données par le générateur en différentes classes.

Pour plus de détail, lire les commentaires dans le code

### Reproduire l’étude avec des nombres aléatoire générés par la fonction rand

Nous avons à présent réalisé la même étude que précédemment mais cette fois-ci avec la fonction « rand » déjà existante sous Scilab.

Plus de détails sont dans les commentaires du code.

### Comparer et Conclure

On remarque que, malgré que ce soit 2 fonctions génératrices de nombres aléatoires, le test du Chi2 est plus souvent vraie pour la loi Uniforme qu’avec le ‘rand’ de Scilab. En effet, la fonction ‘grand’ produit des séquences de nombres qui possèdent de meilleures qualités statistiques que ‘rand’.

## Partie 2 :

### Construire des séquences de nombres d’une loi discrète et continue choisies. Les représenter graphiquement.

A partir des nombres générés en partie 1, nous devions pouvoir construire une séquence de nombres aléatoires de loi discrète ou continue.

Pour cela, plusieurs méthodes sont disponibles :

- La méthode du rejet

- La génération par la fonction de répartition

- La génération par changement de variable

Plus d’informations sont en commentaires au sein du programme.

### Utiliser le test du Chi 2 pour caractériser les séquences générées.

Plus d’informations sont dans les commentaires du programme.

## Partie 3 :

### Utiliser la méthode de rejet pour simuler la loi continue choisie auparavant

Les informations sont commentées dans le programme.

# Exercice 2 : Marche aléatoire

## Partie 1 :

### Donner la loi de Xi.

La probabilité d’avoir pile ou non c’est-à-dire de se déplacer à gauche ou pas est une expérience de Bernoulli de paramètre.

Ici le succès est d’avoir pile.

### Donner la loi X qui représente la position du marcheur à l’instant nT.

On a n expérience de Bernoulli indépendantes. La variable aléatoire X qui compte le nombre de succès parmi n expériences suit la loi binomiale B(nT,).

La probabilité d’obtenir 0 « pile » est :

\*

Donc la probabilité d’avoir au moins 1 « pile » est :

\*

### Représenter les trajectoires de X pour n=20, 100, 1000.

Plus de détails sont dans les commentaires du programme.

### Calculer l’espérance et la variance de X

On a: X (nT, ω) = X1 + X2 + … + Xn

On peut déterminer la position après n itérations, en prenant la valeur 0 pour la marche initiale, en ajoutant 1 pour chaque pas à gauche (pile), en retranchant 1 pour chaque pas à droite (face). Par rapport à la loi binomiale classique il suffit donc de décaler les résultats de et de multiplier par 2, ainsi :

L’espérance de X:

E(X) =

La variance de X :

V(X) = E(X2) - E(X) 2

Avec :

E(X2) =

Donc :

V(X) = n – 0 = n

## Partie 2 :

### Expliquer pourquoi on ne prend pas s=αT

### Représenter des trajectoires de X pour n fixe pour T de plus en plus petit.

Voir le programme Scilab.

### Démontrer que X(t) suit la loi N(0,).appelée le mouvement Brownien

## Partie 3 :

### Refaire la partie 1) en deux dimensions

Voir le programme sur Scilab.