Université de Nice
Département de Mathématiques

Année 2005-2006
Licence MI/MP 2e année

Statistiques : Enoncé du TP 2
Méthode de Monté Carlo : estimation de la valeur de π

1. Simulations de nombres aléatoires :

1. La fonction `rand()` de Scilab (comme la touche `random` d'une calculette) renvoit un nombre aléatoire compris entre 0 et 1, distribué selon une loi uniforme sur $[0, 1]$. Si l'on précise `rand(m,n)`, on obtient une matrice de taille $m \times n$ dont toutes les composantes sont des nombres aléatoires distribués selon une loi uniforme sur $[0, 1]$. Faire quelques essais pour diverses (petites) valeurs de m et de n.
2. La suite des instructions suivantes permet de vérifier si la loi simulée est bien une loi uniforme : `x=rand(1,1000);histplot(10,x)` Mettre un titre à la figure obtenue et la sauvegarder dans l'éditeur de votre choix, en haut à gauche d'une feuille que vous imprimerez en fin de séance. **Taper vos nom et prénom en haut de cette feuille.**
3. La fonction `2*rand()-1` renvoit encore un nombre aléatoire, mais il n'est plus distribué selon une loi uniforme sur $[0, 1]$. Faire quelques essais puis déterminer quelle loi est simulée par cette fonction.
4. Taper la fonction `rand(1,1,'normal')` et l'exécuter plusieurs fois. Puis, sur le modèle de la question précédente, simuler un vecteur ligne g de 1000 composantes, distribuées selon une loi normale centrée réduite et tracer son histogramme (en choisissant également 10 classes). Mettre un titre et sauvegarder cette nouvelle figure (qui comporte, à présent, deux histogrammes superposés) à coté de la précédente.

2. Simulation de points aléatoires dans un carré :

On désigne par X et Y deux vecteurs ligne de 1000 composantes simulants chacun 1000 nombres aléatoires uniformément répartis dans l'intervalle $[-1, 1]$. Créer ces deux vecteurs. Vérifier en tapant les instructions `xset("window",1);plot2d(X,Y,-4)` qu'on peut ainsi simuler 1000 points uniformément répartis dans le carré $[-1, 1] \times [-1, 1]$. Qu'advient-il si l'on change l'argument optionnel -4 en une autre valeur ? Ne pas sauvegarder cette figure (on la fermera par exemple en cliquant sur la croix en haut à droite de la fenêtre graphique)

3 Distinguer les points du disque des autres :

Pour représenter de façon différente les points qui sont dans le disque D d'équation $X^2 + Y^2 < 1$ et ceux qui n'y sont pas, on introduit un nouveau vecteur ligne Z dont la i-ième composante vaut 1 ou 0 selon que le point de coordonnées $X(i), Y(i)$ est dans le disque ou non. On peut utiliser pour cela l'instruction

```
Z=bool2s(X^2+Y^2<1)
```

Pour obtenir ensuite le dessin recherché, il suffit de taper :

```
xset("window",1);for i=1 :1000,plot2d(X(i),Y(i),-2*Z(i)-2),end
```

Sauvegarder cette figure (après y avoir ajouté un titre) sur la feuille qui en comporte dejà deux.

4. Estimations du nombre π :

Le vecteur Z va également servir à compter le nombre de points, parmi les 1000 points simulés, qui appartiennent au disque D. Il suffit pour cela de taper `sum(Z)`.
En déduire une estimation du nombre π.
Relancer plusieurs fois le calcul de X et Y afin d'obtenir d'autres simulations de 1000 points du carré, et en déduire d'autres estimations de π (formule vue en cours).

5. Propriétés de l'estimateur de π :

1. Afin d'étudier les propriétés de cet estimateur de π, on construit un vecteur noté Pi dont les 400 composantes sont 400 simulations successives de π, au moyen de la suite des instructions suivantes
```
for j=1 :400, X=2*rand(1,1000)-1, Y=2*rand(1,1000)-1,
Z=bool2s(X^2+Y^2<1), Pi(j)=4*sum(Z)/1000, end
```
2. Par l'instruction `histplot(20,Pi)`, on peut tracer l'histogramme. Calculer la moyenne empirique (`mean`), l'écart type empirique (`stdev`) puis l'étendue (avec `min` et `max`) de cette suite d'estimations de π et retrouver ces quantités sur votre histogramme.
3. On va comparer cette distribution empirique avec la distribution théorique que l'on obtiendrait à la limite lorsque le nombre d'estimations de π calculées (ici 400) tend vers l'infini. Il s'agit de la loi normale $\mathcal{N}(m, \sigma)$ dont la densité est donnée par la formule
$$N(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$
où m et σ sont les espérance et écart type théoriques de cet estimateur de π (dont les expression ont été calculées dans le cours). Définir cette fonction N en utilisant la syntaxe
```
function y=N(x) ;y=.... ;endfunction
```
Controler en calculant quelques valeurs remarquables de N.
4. Sans changer la fenêtre graphique, superposer le graphe de cette fonction avec l'histogramme précédent par
```
fplot2d(min(Pi) :0.01 :max(Pi),N)
```
Mettre un titre à ce graphique et le sauvegarder avec les 3 premiers sur une feuille **portant vos noms et prénoms** que l'on imprimera (en prenant soin de contrôler, **avant l'impression**, que l'ensemble des 4 graphiques tiennent bien **sur une seule feuille**).

6. Contrôle de la règle des 95% (et de celle des 68%) :

Nous avons vu que l'une des plus importantes propriétés de la loi normale est qu'elle prend environs 95% de ses valeurs dans l'intervalle $I_{2\sigma} = [m-2\sigma, m+2\sigma]$ (et environs 68% de ses valeurs dans l'intervalle $I_\sigma = [m-\sigma, m+\sigma]$). Cela signifie que, en théorie, on devrait avoir ici environs 380 parmi les 400 simulations de π réalisées qui appartiennent à l'intervalle $I_{2\sigma}$ et environs 272 qui appartiennent à I_σ. Calculer les extrémités de ces deux intervalles puis, en vous inspirant de la définition du vecteur Z, définir un vecteur de 400 composantes, égales à 0 ou 1 selon que l'estimation de π est ou n'est pas dans l'intervalle $I_{2\sigma}$ et faire de même pour I_σ. La somme des composantes de ces vecteurs donnera le nombre d'estimations appartenant à l'un puis l'autre de ces deux intervalles.

7. Influence de la taille de l'échantillon sur la qualité de l'estimateur :

Reprendre l'ensemble des questions 5 et 6 en remplaçant le nombre de simulations (1000) par 10000 et comparer les propriétés du nouvel estimateur (attention, son ecart type théorique n'est plus le même).