Module Probabilités et applications
Travaux pratiques de simulation

Michel DE LARA, Benjamin JOURDAIN, Tony LELIEVRE

December 7, 2017

Contents

La séance de TP se fait sous environnement Linux. Pour commencer la séance, ouvrir un shell, créer un répertoire 'tp_scilab' (`mkdir tp_scilab`) puis se placer sous ce répertoire (`cd tp_scilab`). Lancer un navigateur, par exemple Mozilla (`mozilla &`), et aller sur la page web suivante :

```
http://cermics.enpc.fr/scilab_new/site/Tp/Probabilites/tp_proba_Q
```

Lancer ensuite Scilab depuis ce répertoire 'tp_scilab' (`scilab`) et ouvrir une fenêtre d'un éditeur, par exemple emacs (`emacs &`).

Pour chacune des questions, vous pouvez soit utiliser un "copier-coller", soit télécharger un programme `Q_i.sce` ($i = 1..8$) qui contient l'essentiel des commandes, avec quelques lignes à compléter. Pour télécharger le programme `Q_i.sce`, utiliser par exemple le bouton droit de la souris et la commande 'Save Link Target as' puis choisissez le répertoire 'tp_scilab' et le nom de fichier `Q_i.sce`. Il suffit ensuite d'éditer ce programme (par exemple avec emacs) pour le modifier. Pour l'éxécuter, il suffit de taper sur la ligne de commande Scilab :

```
-->exec Q_i.sce;
```

Pour un rappel sur les opérations élémentaires de Scilab, on renvoie à l'*Introduction générale à Scilab pour les travaux pratiques à l'*ENPC :

Introduction à Scilab / Manipulations vectorielles

Introduction à Scilab / Graphiques, fonctions Scilab, programmation, saisie de données

On rappelle que pour avoir de l'aide sur une commande Scilab, il suffit de taper sur la ligne de commande Scilab :

```
-->help nom_de_la_commande;
```

1 Test du générateur aléatoire

Nous allons utiliser le générateur aléatoire de Scilab `rand`. On rappelle que la commande `rand(n,m)` renvoie une matrice aléatoire de taille $n \times m$, dont les composantes sont des réalisations indépendantes de variables aléatoires de loi uniforme sur $(0,1)$.

Pour le tracé d'histogrammes, nous allons utiliser la commande `histplot`. On rappelle que la commande `histplot(n,x)` trace un histogramme des valeurs contenues dans le vecteur x, avec n barres de même largeur. On peut également utiliser la commande `histplot(r,x)`, avec r un tableau donnant les valeurs pour échantillonner x.

Question 1 *Générer un vecteur de taille $N = 1000$ (i.e. une matrice de dimension $(1,N)$) dont les composantes sont des réalisations indépendantes de variables aléatoires de loi uniforme sur $(0,1)$ avec la fonction `rand`. Tracer l'histogramme correspondant avec la fonction `histplot`. Augmenter N ($N = 10\ 000, 100\ 000$...). Que constatez-vous ? Pouvez-vous l'expliquer ?*

Télécharger Q_1.sce

```
// Remplacer les ... par la commande appropriée
N=1000;// Nombre de réalisations de la variable aléatoire générée
// Génère un tableau x de taille (1,N) de réalisations
// de variables aléatoires indépendantes uniformes sur (0,1)
 ...
xbasc();// Efface la fenêtre graphique
 ...
// Trace un histogramme de x avec 100 barres.
```

2 Simulation de variables aléatoires de loi exponentielle

Question 2 *Choisir un réel $\lambda > 0$ et un entier N assez grand ($100, 1\ 000, 10\ 000$...). Écrire une ligne de code Scilab qui retourne N réalisations indépendantes de loi exponentielle de paramètre λ. Tracer l'histogramme de ces N réalisations et lui superposer la densité de la* *loi exponentielle. Vérifier graphiquement la proximité à la loi originale. (On rappelle qu'en Scilab, si x est un vecteur, alors $\log(x)$ est le vecteur de composantes le logarithme des composantes de x.)*

Télécharger Q_2.sce

```
// Remplacer les ... par la commande appropriée
lambda=0.5;// Paramètre de la loi exponentielle
N=10000;// Nombre de réalisations de la variable aléatoire générée
// Génère un tableau z de taille (1,N) de réalisations d'une variable
// aléatoire exponentielle de paramètre lambda
 ...
xbasc();
// Tableau de discrétisation des abscisses
// L'intervalle (0,12) est divisé en 24 intervalles de même longueur
x=linspace(0,12,25);
 ... // Trace l'histogramme de z échantillonné suivant x
// Calcule y, l'image de x par la fonction densité
// de la loi exponentielle de paramètre lambda
 ...
// Trace y en fonction de x
plot2d(x,y);
```

3 Simulation de variables aléatoires de loi binomiale

Question 3 *Expliquer pourquoi le vecteur y du programme suivant renvoie bien un vecteur de taille N dont les composantes sont des réalisations d'une variable aléatoire de loi binomiale de paramètres (n, p). Faire varier p et commenter les résultats.*

Cette méthode pour générer des réalisations d'une variable aléatoire de loi binomiale est préférable à l'utilisation de boucles `for` *qui sont très lentes en Scilab.*

On rappelle que la fonction `sum(w,'r')` *appliquée à une matrice w de taille $n \times m$ renvoie une matrice de taille $1 \times m$ dont chaque composante contient la somme des composantes de w par colonnes.*

Télécharger Q_3.sce

```
n=10;p=0.4;// Paramètres de la loi binomiale
N=10000;// Nombre de réalisations de la variable aléatoire générée
// Génère un tableau y de taille (1,N) de réalisations
// d'une variable aléatoire binomiale de paramètres (n,p)
y=sum(rand(n,N) < p,'r');
xbasc();
// Trace le diagramme en bâton de y
```

```
[ind,occ]=dsearch(y,0:n,"d");
xbasc();plot2d3(0:n,occ/N,rect = [-1,0,n+1,max(occ/N)*1.1],nax = [0,n+3,1,9]);

// Trace en bleu les probabilités exactes
plot2d3((0:n)+0.2,binomial(p,n),style = 2);
```

4 Loi forte des grands nombres

Question 4 *Tirer N réalisations indépendantes $X_1,\ldots,X_N$ d'une loi exponentielle et tracer le graphique donnant $k \mapsto \dfrac{X_1+\cdots+X_k}{k}$. Commenter.*

On utilisera la division terme à terme `./` *et la fonction* `cumsum` *qui évalue des sommes cumulées, plutôt qu'une boucle. On rappelle par ailleurs que l'on peut générer le vecteur $(1,2,...,N)$ sous Scilab par* `(1:N)`.

Télécharger Q_4.sce

```
// Remplacer les ... par la commande appropriée
lambda=5;// Paramètre de la loi exponentielle
N=10000;// Nombre de réalisations de la variable aléatoire générée
// Génère un tableau x de taille (1,N) de réalisations d'une variable
// aléatoire exponentielle de paramètre lambda
 ...
xbasc();
// Crée un tableau y de taille (1,N) tel que y(i)=(x(1)+...+x(i))/i
 ...
// Trace y
plot(y);
```

Question 5 *Même question avec des X_i i.i.d. suivant une loi de Cauchy de paramètre a. Exécuter plusieurs fois le programme et commenter.*

On rappelle que sous Scilab, la valeur de π est stockée dans `%pi`.

Télécharger Q_5.sce

```
// Remplacer les ... par la commande appropriée
a=5;// Paramètre de la loi de Cauchy
N=1000;// Nombre de réalisations de la variable aléatoire générée
// Génère un tableau x de taille (1,N) de réalisations d'une variable
// aléatoire de Cauchy de paramètre a
 ...
xbasc();
// Crée un tableau y de taille (1,N) tel que y(i)=(x(1)+...+x(i))/i
 ...
// Trace y
plot(y);
```

5 Théorème de la limite centrale

Question 6 *Choisir un entier n assez grand. Tirer n réalisations indépendantes $X_1,\dots,$ X_n d'une loi uniforme sur $[0,1]$ et calculer $\dfrac{X_1+\cdots+X_n-n/2}{\sqrt{n/12}}$. Choisir un entier N assez grand. Recommencer N fois et tracer un histogramme de la loi de $\dfrac{X_1+\cdots+X_n-n/2}{\sqrt{n/12}}$. Superposer la densité d'une loi normale centrée réduite. Jouer sur les paramètres n et N. Commenter.*

On pourra utiliser la fonction `sum(.,'r')`*).*

Télécharger Q_6.sce

```
// Remplacer les ... par la commande appropriée
// stacksize(10000000);
// A décommenter si problème de mémoire ('stack size exceeded')
n=50;// indice pour la convergence du TCL
N=1000;// nombre de réalisations pour tracer l'histogramme
// Crée un tableau w de taille (n,N) contenant des réalisations
// indépendantes d'une variable aléatoire uniforme sur (0,1)
 ...
// Calcule le vecteur z de taille (1,N) tel que
// z(i)=((w(1,i)+...+w(n,i))-n/2)/sqrt(n/12)
 ...
xbasc();
// Tableau de discrétisation des abscisses
// L'intervalle (-3,3) est divisé en 24 intervalles de même longueur
x=linspace(-3,3,25);
 ... // Trace l'histogramme de z échantillonné suivant x
// Raffine la grille en abscisse pour tracer la densité
x=linspace(-3,3,250);
// Calcule y, l'image de x par la fonction densité
// de la loi normale centrée réduite
 ...
// Trace y en fonction de x
plot2d(x,y);
```

6 Simulation de variables aléatoires de loi bêta (2,2) par la méthode du rejet

La variable aléatoire X suit une loi *bêta* de paramètres $(2,2)$ si X admet la densité

$$p(x) = 6x(1-x)\mathbf{1}_{[0,1]}(x).$$

On propose de simuler cette variable aléatoire par la méthode du rejet en comparant cette loi à la loi uniforme sur $(0,1)$. On a évidemment :

$$6x(1-x)\mathbf{1}_{[0,1]}(x) \leq (3/2)\mathbf{1}_{[0,1]}(x).$$

On sait donc que si on se donne une suite i.i.d. (Y_i, U_i) avec Y_1 et U_1 indépendants de loi uniforme sur $(0,1)$, alors X a même loi que Y_N où $N = \inf\{i,\ (3/2)U_i \leq p(Y_i)\}$.

Question 7 *Programmer une fonction qui rend un vecteur de N réalisations indépendantes suivant la loi* bêta *de paramètres* $(2,2)$ *(on pourra utiliser une boucle* `while` *: taper* `help while` *pour avoir une description de la commande). Vérifier que l'on obtient bien la bonne densité en traçant un histogramme.*

Télécharger Q_7.sce

```
// Remplacer les ... par la commande appropriée
function x=rejetbeta(N)
  x=zeros(1,N);
  for i=1:N do
    // Mettre une réalisation d'une variable aléatoire
    // suivant une loi beta de paramètre (2,2) dans x(i).
     ...
  end;
endfunction
N=10000;// Nombre de réalisations de la variable aléatoire générée
z=rejetbeta(N);
// Tableau de discrétisation des abscisses
// L'intervalle (0,1) est divisé en 24 intervalles de même longueur
x=linspace(0,1,25);
xbasc();
histplot(x,z);// Trace l'histogramme de z échantillonné suivant x
// Calcule y, l'image de x par la fonction densité
// de la loi beta de paramètre (2,2)
y=6*x .*(1-x) .*(x < 1 & x > 0);
// Trace y en fonction de x
plot2d(x,y);
```

7 Simulation de variables aléatoires de loi normale

Question 8 *Choisir un entier N assez grand* ($100, 1\ 000, 10\ 000$...)*. Utiliser la méthode polaire pour créer un vecteur de taille N dont les composantes sont des réalisations indépendantes d'une loi normale centrée réduite. Tracer l'histogramme de ces N réalisations et lui superposer la densité de la loi normale. Vérifier graphiquement la proximité à la loi originale. On pourra utiliser la multiplication terme à terme* `.*`.

Télécharger Q_8.sce

```
// Remplacer les ... par la commande appropriée
N=1000;// Nombre de réalisations de la variable aléatoire générée
// Génère un tableau z de taille (1,N) de réalisations d'une variable
// aléatoire normale centrée réduite
 ...
xbasc();
// Tableau de discrétisation des abscisses
// L'intervalle (-3,3) est divisé en 24 intervalles de même longueur
x=linspace(-3,3,25);
 ... // Trace l'histogramme de z échantillonné suivant x
// Raffine la grille en abscisse pour tracer la densité
x=linspace(-3,3,250);
// Calcule y, l'image de x par la fonction densité
// de la loi normale centrée réduite
 ...
// Trace y en fonction de x
plot2d(x,y);
```