ECS1.1 - TP N ° 5 : Graphiques et simulations aléatoires en Scilab

Exercice 1 Représentation de la courbe d'une fonction numérique

Si `x` et `y` sont deux vecteurs de même dimension, la commande `plot2d(x,y)` dessine une ligne brisée entre les points dont les abscisses sont données par le vecteur `x` et les ordonnées par le vecteur `y`.
Pour subdiviser un intervalle $[a, b]$ en n points (donc $(n-1)$ sous-intervalles), on dispose des instructions `x=linspace(a,b,n)` ou `x=a:(b-a)/(n-1):b` .

1. Comparer le résultat des instructions :

```
linspace(0,1,11)                    1:0.1:10
```

Créer un vecteur ligne `x` contenant une subdivision de $[-2\pi, 2\pi]$ en 1000 points (on rappelle que `%pi` donne en Scilab une valeur approchée de π). Quel est l'intérêt de l'inhibiteur d'affichage `;` ?

2. Comparer le résultat des instructions :

```
z=linspace(0,%pi,10);
y=zeros(1,10)
for i=1:10
     y(i)=cos(z(i));
end
```

```
z=linspace(0,%pi,10);
y=zeros(1,10);
y=cos(z);
```

Bon à savoir : l'instruction `y=zeros(1000,1)` sert à initialiser la matrice `y` avant son affectation et accélère significativement le temps d'exécution du programme. Mais elle est facultative.
Remarquer qu'on peut appliquer la fonction `cos` à une matrice.
Créer un vecteur ligne `y` dont les composantes sont égales à celles de `x` auxquelles on a appliqué la fonction cosinus.

3. Exécuter l'instruction `plot2d(x,y)`. Elle permet de visualiser la courbe de la fonction cos sur $[-2\pi, 2\pi]$. A l'aide du menu Edition de la fenêtre figure, essayez de changer la couleur de la courbe, de donner un titre à la figure, de mettre une légende sur les axes.

4. Si `x` est un vecteur et `y` une matrice dont le nombre de colonnes est égal à la dimension de `x`, la commande `plot2d(x,y)` dessine des lignes brisées entre les points dont les abscisses sont données par le vecteur `x` et les ordonnées par chaque colonne du vecteur `y` (on obtient autant de lignes brisées que le nombre de colonnes de la matrice `y`).

Fermer la figure puis exécuter les instructions (qu'elle est la différence entre `^` et `.^` ?) :

```
--> x=linspace(0,4,501);

--> y1=x.^2; y2=exp(x); y3=x.^3;

--> plot2d(x',[y1',y2',y3'])
```

Remarques : il est cette fois important d'avoir des vecteurs colonnes, c'est pourquoi on utilise la transposée. La commande `[y1',y2',y3']` correspond à une concaténation horizontale de vecteur colonne (cf cours).

Exercice 2 Représentation des termes d'une suite : application à la ruine du joueur

Si `u` est un vecteur contenant n termes consécutifs d'une suite, on peut les visualiser graphiquement grâce à l'instruction `plot2d(1:n,u)` .

1. Représenter les n premiers termes de la suite $\left(\frac{(-1)^n}{n}\right)_{n\in\mathbb{N}^*}$, pour différentes valeurs de n.
Pour créer le vecteur `u` on n'utilisera pas de boucle `for`, mais seulement les opérations matricielles coefficient par coefficient.

2.a) A l'aide de l'instruction `rand()`, donner une fonction `piec(p)` qui simule le lancer d'une pièce qui donne « pile » avec probabilité p.

2.b) On considère un joueur disposant d'une fortune de a euros jouant contre un casino disposant d'une fortune de b euros.
A chaque tour il gagne un euro avec probabilité p et perd un euro avec probabilité $1-p$.
Le jeu s'arrête lorsque le joueur ou le casino est ruiné. Nous avons démontré en TD de mathématiques que ceci se produit de manière presque sûre.
Que représente le vecteur `fortune` à la fin de la fonction suivante ?

```
function[]=ruine(a,b,p)
fortune=[a];
n=0;
while fortune(n+1)<>0 & fortune(n+1)<>a+b
    n=n+1;
    if rand()<p then fortune(n+1)=fortune(n)+1;
                else fortune(n+1)=fortune(n)-1;
    end
end
endfunction
```

La compléter pour qu'en sortie elle représente graphiquement l'évolution de la fortune du joueur jusqu'à la fin du jeu et donne le nombre de tours de jeu effectués. Penser à fermer la figure avant chaque nouvel appel de la fonction (ou a rajouter l'instruction `clf` avant l'instruction `plot2d`) !

2.c) Modifier cette fonction pour qu'elle ne produise plus de figure, et donne le temps moyen de jeu calculé sur 1000 simulations différentes. Comparer avec la valeur théorique :

$$T = \begin{cases} \dfrac{1}{p-q}\left(\dfrac{(a+b)(x^a-1)}{x^{a+b}-1} - a\right) & \text{si } p \neq \dfrac{1}{2} \\ ab & \text{si } p = \dfrac{1}{2} \end{cases}$$

où $q = 1-p$ et $x = \dfrac{q}{p}$.
<u>Astuce Scilab</u> : pour calculer la somme des coefficients du vecteur *ligne* `V` sans boucle `for`, il suffit de le multiplier par le vecteur *colonne* constitué uniquement de 1, de même taille. En divisant ensuite par la taille, on obtient la moyenne des coefficients de `V`.

Exercice 3 Simulations d'une loi uniforme ou d'une loi binomiale - Théorème de Moivre Laplace

1.a) A l'aide de l'instruction `rand()`, simuler une réalisation d'une variable de loi uniforme sur $[\![1,10]\!]$. Proposer une version avec boucle `for` et une autre sans (grâce à l'instruction `find`, voir cours).

Autre solution plus simple : pour simuler une matrice X $r\times s$ aléatoire, et dont chaque coefficient est choisi de manière indépendante selon la loi uniforme sur $[\![a,b]\!]$ on dispose de l'instruction `X=grand(r,s,'uin',a,b)`.

Si c et V sont deux vecteurs lignes, la commande `histplot(c,V)` compte le nombre de coefficients de `V` dans les intervalles :

$$[c(1), c(2)] \quad]c(2), c(3)] \quad \ldots]c(k), c(k+1)] \ldots$$

(attention aux bornes des intervalles) et trace l'histogramme de ces nombres (normalisé d'aire égale à 1).

1.b) A l'aide de l'instruction `grand`, créer un vecteur ligne X correspondant à 1000 simulations indépendantes de la loi uniforme sur $[\![1, 10]\!]$.
Exécuter les instructions suivantes : `c=-0.5:1:10.5; histplot(c,X).`
Que représente cette figure ? Recommencer avec un nombre de simulations plus grand ou plus petit. Théoriquement à quoi doit ressembler cette histogramme ?

L'instruction `plot2d3` permet de tracer un diagramme en bâtons. Exécuter les instructions :

```
--> c=-0.5:1:10.5;
--> histplot(c,X)
--> plot2d3(1:10,0.1*ones(1,10),5) // diagramme en bâtons de la loi uniforme sur 1,...,10
// le 5 sert à imposer la couleur rouge
```

2.a) A l'aide de l'instruction `rand()` et d'une boucle for, donner une fonction `binomiale(n,p)` qui donne en sortie une réalisation de la loi binomiale (n, p). Proposer aussi une version sans boucle `for` utilisant l'instruction `find()` (voir cours).

Autre solution plus simple : on dispose dans Scilab de l'instruction `X=grand(r,s,'bin',n,p)` qui crée une matrice $r \times s$ aléatoire dont chaque coefficient est choisi de manière indépendante selon la loi binomiale (n, p).

2.b) A l'aide de l'instruction `grand`, créer un vecteur ligne X correspondant à 1000 simulations indépendantes de la binomiale (n, p) pour différentes valeurs de n et p.
Exécuter les instructions suivantes : `c=-0.5:1:(n+0.5); histplot(c,X).`
Que représente cette figure ? Recommencer avec un nombre de simulations plus grand ou plus petit.

Pour comparer avec le diagramme en bâtons théorique, exécuter les instructions :

```
--> X=grand(1,1000,'bin',10,0.33);
--> c=-0.5:1:10.5;
--> histplot(c,X)
--> plot2d3(0:10,binomial(0.33,10),5) // diagramme en bâtons de la loi binomiale (10,0.33)
// le 5 sert à imposer la couleur rouge
```

2.c) On rappelle le théorème de Moivre-Laplace. Si $X_n \hookrightarrow \mathcal{B}(n, p)$:

$$\forall (a, b) \in \mathbb{R}^2 \text{ avec } a < b, \quad \lim_{n \to +\infty} \mathbb{P}\left(a \leq \frac{X_n - np}{\sqrt{np(1-p)}} \leq b\right) = \frac{1}{\sqrt{2\pi}} \int_a^b \mathrm{e}^{-\frac{t^2}{2}}\, dt$$

Si on pose $Z_n = \dfrac{X_n - np}{\sqrt{np(1-p)}}$ on peut donc, pour de grandes valeurs de n, approximer la loi de Z_n par la loi de Gauss $\mathcal{N}(0, 1)$.

Exécuter les instructions :

```
--> X=grand(1,100000,'bin',5000,0.33); // 100 000 simulations de B( 5000 , 0.33 )
--> Z=(X-5000*0.33)/sqrt(5000*0.33*(1-0.33));
--> c=-4:0.2:4;  // vecteur des abscisses
--> histplot(c,Z);  // histogramme des valeurs de Z
--> Y=1/sqrt(2*%pi)*exp(-c.^2/2);   // densité de la loi de Gauss N(0,1)
--> plot2d(c,Y)  // représentation de la courbe de cette densité
```

Comparer graphiquement la loi de Z_n et la loi de Gauss.